Title	Game Recommendation Base On Deep Clustering
Author(s)	常, 濮俊
Citation	
Issue Date	2020-03
Type	Thesis or Dissertation
Text version	author
URL	http://hdl.handle.net/10119/16404
Rights	
Description	Supervisor:Dam Hieu Chi, 先端科学技術研究科, 修士(情報科学)

Master's Thesis

Game Recommendation Base On Deep Clustering

Chang Pujun

Supervisor : Dam Hieu Chi

Graduate School of Advanced Science and Technology
Japan Advanced Institute of Science and Technology
(Information Science)

February, 2020

Abstract

Information overload widely exists after the Internet age came. As the entertainment industry booming, when users are facing massive game resources, the information overload problem was brought to the forefront under the circumstance. So far, the research concerning game recommendation most relies on the users-Profile based recommendation, like games tags from users, or users chronological behavior or history, or user collaboration recommendation, etc. However, the similarity calculation base on the game itself is absent. Therefore, in this research, we will dig more on the attribution of games itself, and evaluate the performance of the recommendation system, which calculate by the new representations from data.

In this study, we acquired 4000 game fragments, attempt to build one Recommendation System to improve the accuracy of game recommendation. The experiment tried to explore a potential method to measure the distance between various articles under content-based recommendation.

The whole Recommendation System in this research base on the content-based recommendation. Typically, the content-based recommendation has a positive consequence. Moreover, it would not face the "content-cold start" problem. There are two sections in this recommendation structure. The first data management section would use frames from games to trained one deep learning neural network for visual representations extraction and build recommendation database. Then aim to speed up the recommendation part, the research would use a clustering method to manage data into a recommendation database. In the second recommendation section, the well-trained model would use to extract the input of users which is the games playing by user in real time. Then the recommendation system would calculate top-N candidates for recommendation list according to the input of users.

The experimentation result indicated a stable consequence of the Recommendation System. Most recommendation lists demonstrate that the Recommendation System would capture the visual representations of frames precisely from users' input frames with quick response. The research described a solution for solving the information overload issue in-game areas. It took the frames and scenes from games as a unit, instead of the conventional way of whole games as a unit for various games' similarity calculation. The research proved the practicability of the Recommendation System from the visual impact. Furthermore, it uses a deep learning method for exploring the new data representations from games to measure the similarity between various games. The research could effectively enhance user stickiness and create more value for the game platform.

Keywords: Game Recommendation, information overload, Deep Learning, Clustering

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

The game industry nowadays, It is showing a rapid increase in this business. According to the U.S. game market report by Nielsen's professional game data analytics company, after analysis of the game industry, which based on U.S. Census data in 2018. Two-Thirds of the U.S. population 13 years and older are gamers, up from 58% in 2013. Also, the scale of self-identified gamers had a steady growth. The analysts gave an optimistic prediction about 118.2 billion revenue of the game industry in the next coming year. Also, according to the report, online watching game content becoming a new trend in recent years. With millions of viewers tuning in to both live-streamed gaming and pre-recorded videos. On the other side, over the ocean, according to the Chinese Carton and games development report, the market growing from 1144 hundred million in the year 2014 to 2144 hundred million in the year 2018. Furthermore, it gave a prediction of a positive future of the game industry of China.

Meanwhile, the rapid growth of the games industry stimulates the scale of games development. Therefore, it also made the information overload problem of the game industry inevitable. Individuals were submerging in different kinds of games, which the situation makes gamers disoriented and hard to make a good choice. On the other hand, even for the game makers, it is becoming more arduous by traditional advertising method to capture users or increasing the market profit of new games. Moreover, the Sense of game experience demand by game users upgraded during the rapid growth of game industry, the willingness of gamers to choose games becoming strong. In a nutshell, although the data is showing a hopeful expansion of the global game market. However, all these changes still might lead to uncertainty in the future.

1.2 Related Work

Individuals have been put effort into three approaches to give solutions to the information overload problem[2]. Category Navigation, Search Engine, and Recommendation System. Category Navigation tries to track the interfaces of online resources, then integrated resources and give categorize tags to all sources. Users could go through the available online resource from the navigation page provided by the web editor. Above all, Category would need plenty of extra artificial maintenance. Also, the effectiveness of Category Navigation limits its growth. Therefore, the Category Navigation has been replacing by the Search Engine. Search Engine using a web crawler to crawl the webpage automatically, after that build index for these webpages by specific rules and storage in the local environment. Then the ranking algorithm sort all webpages based on the correlation between the critical word defined by users and all webpages. Comparing to Category Navigation, Search Engines save extra artificial work for maintenance and categorization.

Nevertheless, the Search Engine requires one keyword defined by users. Considering sometimes users have no clue about describing the interest area, That might seriously affect the effect when applying the Search Engine to solve the information overload problem. What is more, there would not be a significant difference when two beginners search the same keyword.

Comparing with Category Navigation and Search Engine, the Recommendation System would be more efficient and smarter. It would be more proactive in information push and recommendation base on users‘ individuation, even without the vital description from users’ needs. For the first time, David Goldberg, David Nichols, Brain Oki, and Douglas Terry use Collaborative Filtering thinking to apply the Tapestry mail filtering system in 1992[3], the information management enter the new epoch of recommendation system . Then in 1995, the first company, named Agent commercialize the recommendation system. Until Association for Computing Machinery organized the first top recommendation conference ReeSys in 2007. The recommendation system attracting more and more eyes, SIGIR (Special Interest Group on Information Retrieval), KDD (Knowledge Discovery in Database), and so many top conferences remain specialized conferences for recommendation system topics.

The personalization recommendation system used to settle the information overload problem at the beginning. It recommends items or sources according to the users’ interests or backgrounds, especially when facing tremendous data. Quite a few giant online service companies like Amazon, Google, Baidu, researchers spent plenty of time to develop personalization recommendation system to help users to make decisions. As an example, Youtube made a great effort in its personalization recommendation system. By the paper which published by Google developer Pail Covington, Jay Adams and Emre Sargin in 2016[4].

The whole recommendation system made up of two parts, a deep candidate generation model using the representations data of users ' history under collaborative filtering to select candidates from the whole database. The Ranking network will grade each candidate according to the objective function, which includes two parameters, contents, and history of users with features of the video. The recommendation algorithm significantly promotes the users watching time of steam under the A/B test.

So as one vital area of recommendation system, the game recommendation is becoming a feasible way to help gamers to select one game when they are facing tremendous numbers of games. and be an excellent remedy for information overload in the game market. Prominent game manufacturers like Tencent Games made their own mobile game recommendation system on its Tencent platform, the platform uses users' defined features to analyze, and create more than 200 million active users record and tremendous profit. Online game service website Steam also upgrades its recommendation platform[5] by build one new personalizes a custom page in 2019. It is using a neural network to collect users' history and calculate users' interest. Aim to return initiative to gamers; the games would be recommended by its suitability of users, rather than its popularity.

1.3 Motivation

Although the Vigorous development of game recommendation would be a guarantee to satisfy the increasing users' need also would bring a bright future of game market stimulation. However, there is still a relative lack of research in the game recommendation area.

Although, more aspects are needed to describe more specific attributes, according to most of the largest online game service providers, like popular online game emporium, Steam. The recommendation classifier offers only several options, Number of players, language, operation system, and tags from users. The visual impact which defined by Andy Donovan, Hyerim Cho, Chris Magnifico, Jin Ha Lee[6], the so-called "a cohesive and unifying visual aesthetic" to the games is absence. For modern games, As the visuals of games and its graphic style have become widely diversified, the importance of identifying and categorizing visual information is also increasing. Especially the game is sharing the same VAUDIO (voice+audio) platform[7] with film, television, and animation. Even games are not films, we can still easily infer the connection from more and more games are recompose into films, and vice versa.

As a VAUDIO media, the games provide immersion and visual hallucination by image frames of games, to give gamers artificial visual stimulation. Different games could give different immersion and visual hallucinations to users, also represent the different game styles of games. Like pixel games, trying to decrease the number of lines and nodes, focus on shaping the form with the latest numbers of characters and items, and usually have a substantial world architecture and a high degree of freedom. Perhaps realist style, all aspects of the game graphic focusing on restoring the effects of the real world, reveal a resolute feeling for gamers, and they invariably have some intense fighting scenes including in the games.

Therefore, what if the game recommendation system moves the attention to the game itself. The visual impact is also a vital factor when we analyze the representation data of the games in the recommendation system. We could focus on the visual impact of the games and explore the similarity between games by image frames or graphic style of the games. Then maybe we could improve the accuracy of the recommendation system to give a better recommendation to the gamers who are facing information overload of the game market.

Keywords: Game Recommendation, Deep Learning, Clustering

1.4 Objectives

In this study, we aim to:

> Explore the game recommendation system based on the graphic features of the games. Try to find the new similarity calculation and new connection between games to observe a general insight into the performance of the recommendation system.

1.5 Thesis Outline

The remainder of this thesis is organized as follows:

Chapter2-Literature Review: We provide an overview of technologies about basic Recommendation model. We also introduce some games attributes for similarity calculation in research which used before. The Neural Network of Autoencoder has been mentioned in this chapter.

Chapter3-Methodology: We describe in details the provenance of games data in this research. Moreover, we demonstrate the main idea and approach for processing data. In this chapter, we also write the technologies which use in this Recommendation System.

Chapter4-Experiment: We discuss the data collection method and data pre-processing section. Furthermore, we give detail of model training settings and display some detail and whole work process of Recommendation System.

Chapter5-Evaluation: We discuss several outcome cases from our Recommendation System. Also, we evaluate the stability of our recommendation model.

Chapter6-Conclusion: We evaluate our research, and give suggestion for future work

Chapter 2

Literature Review

2.1 Data Representation in Recommendation System

The data representation are the most vital basis of recommend candidates. As the input of the Recommendation System, Different representations would use for represent various of candidates in different types of recommendation systems. Like using:

User: The service objective of the Recommendation System, also the producer of feedback. Normally refers to individuals

Item: The object, which is the subject for recommending, like the movie, music, etc.

Feedback: The record of behavior from user to subject, Mainly about *explicit feedback*, which could directly express the preference of users, like rates, comments. Second is *implicit feedback*, which could not directly express the preference of users, like browsing history, click behavior or purchase history.

Most of the Recommendation System research is base on the matrix of the tuple data (User, Articles, Comments). However, as the development of Nature Language Processing and Data Analysis and Graphics processing technologies. Other inputs like comments data and more articles attributes are capture the researcher's attention for Recommendation System research.

Under the content-based recommendation system, The attributes like description or tags of candidates, reviews or comments from users are always using as input of Recommendation System for similarity calculation between various of articles. Moreover, using different features between the same pair of articles also would decide different similarity values of articles. Thus, it affects the performance of different recommendation systems. In this section we would give some research about various of representations for similarity calculation in Game Recommendation System

2.1.1 Chronological behavior of users

Under the circumstance that users' preferences change over time, the historical behavior data could reflect one user‘ interest change. Researcher CHEN Yaowang, YAN Wei, YU Dongjin, then collected users' chronological behavior for design game recommendation system[8]. Researchers made one hypothesis that the Chronological behavior representations of users, which calculate from games remain time on users' desktop and last open time and open frequency, have a potential relationship with users' interest and could be the basis for preference prediction in the future.

As they defined users‘ preference P_i for game i could be express by:

$$Pi = \frac{Ni}{CTi - LTi}$$

N represents the total degree of the current game i , CT_i represents the current time, and LT_i represents the last time to play the game i. So, the function turns the preference of one game from a gamer into Quotient of total time and last playing time. Therefore, the heavy preference weight would be given if the total time is large, also considering the time distance between last playing time and current time. This formula reflects the change of users‘ interest. After obtaining the value of users‘ preference to the games, it expresses by the standardized formula:

$$Pi = \frac{Pi - u}{\sigma}$$

u represents the average values of preference P_i from gamer, σ represents the variance.

Then Researchers preprocess users' chronological behavior to vector as the dataset for the input of one deep neural network and use the dataset to train a deep neural network. In the recommendation part, the model uses logistic regression output layers of the deep neural network to give the probability of all games to predict the new gamer's interest. The Quadratic Loss function for prediction evaluation showing below:

$$L = \frac{1}{2}\left\|y - a^L\right\|^2 + \frac{\lambda}{2n}\sum_w^\infty w^2$$

L represents the number of layers in whole neural network topology, a represents the output value of activation.

According to the experiment result, which evaluates by RMSE (Root Mean Square Error) and recalls, with more hidden layers added in the deeper neural network, the score of RMSE and recall are showing a significant rise tendency.

The research shows a reliable and contingency recommendation outcome base on chronological behavior representations of users. It also implies some advanced technologies like deep neural networks could demonstrate excellent performance in the game recommendation area. However, the research limited only took two variances of users' representations, if considering users‘ behavior could be affected by users‘ circumstance and instantaneous temper and so on, to make the representation instability. Like plenty of files exist on the desktop could be interference for users when selecting some games, or unstable of users behavior. That lead to the calculation process of games recommendation would have an unreliable outcome also demonstrate the weak point of the user-based recommendation. Therefore, instead of instability representations, what if we move attention to the game itself instead users in the representation extraction part, since the attribution of articles would not easily change, it might be a potential solution of stable representations picking for the games recommendation system.

2.1.2 Gesture representations in Games

Nowadays, users could be confused if they attempt to select games when facing more than 150,000 IOS games in Apple store and 120,000 Android games in Google Play. And yet, some of the same playing patterns would be shown in extremely high games. Researchers Hao-Tsung Yang, De-Yu Chen, Ya-Xuan Hong, and Kuan-Ta Che suggest a content-based Recommendation system by collecting users‘ touch representations [9]. They made the hypothesis that the touch gesture in the time-sequential of users would represent the play style of the game. As long as the playstyle database of games built, the similarity between the games could be calculated. In this paper, Researchers collect the 94 gesture data of users' touch from 22 games of 5 categories, then mapping gestures of segments into 2-dimensional map by multidimensional scaling (MDS), and calculate the similarity or dissimilarity of different games by cosine distance, finally using Borda count method to calculate the score of candidates and sort all games to do the recommendation. The ratio of recommends game and target game performed in the evaluation part of this research. This research pointing out the relevance of gamers‘ gesture data exists in different games.

Even though the experiment indicated strong relevance between games base on the gesture data of the games, after analyzed the confusion matrix chart, we can still observe some of the recommended games are away from the target games from the resulting chart.

Moreover, if we look deep into the conclusion, then we could deduce the gestures of high similarity in games, also represent operational space and graphics design of high similarity in games. Therefore, the study would have been more useful if we directly cut into the representation picking problem from a visual sphere, then maybe we could get a recommendation list with higher accuracy.

2.1.3 Implicit Feedback of Users

The explicit feedback from users like rating data and comments are widely using in Recommendation System nowadays. However, Given the difficulty of data obtaining or the quality problem of these explicit feedback data in real life, the performance of Recommendation Systems is not always satisfy the researchers. On the contrary, the implicit feedback like click behavior and browse history is easy to catch online.

Therefore, Researchers YU Dong-jin and CHEN Cong made a hypothesis that base on implicit feedback data of user would more suitable for game recommendation[10], because of the high interaction between users and games and the data would not be affect by users prejudice. They collected the play frequency of users and gaming times data $(t^u_{i,k}, d^u_{i,k})$ to build preference model of users. As a result the preference model base on Forgetting Rule Formula $\alpha * t^\beta + \varphi$ is showing below:

$$g_{\alpha,i} = \sum_{k\in K} \left\{ d^*_{i,h} \times [32.76 \times \left(T^*_0 - t^*_{i,h}\right)^{-0.084}] \right\}$$

The $k \in K$ is all record of user u operation to subject game i. $g_{u,i}$ is preference of user u to game i. $d^u_{i,k}$ represent the intensity of operation of k. The left part $32.76 \times \left(T^*_0 - t^*_{i,h}\right)^{-0.084}$ is affection to the judgment of game i from user u, by the operation of k .

Aiming to reach the personalized information of users, The LOSS function in this research is not only including baseline score, also do decompose the users preference matrix to users eigenmatrix Q and games eigenmatrix P. After add the personalize factor, the predict preference of user u to game i is:

$$= \mu + b_i + q_i^{\mathrm{T}} \cdot \left(p_u + \frac{\sum_{i=1}^{m_u} q_i}{\sqrt{m_u}}\right)$$

In the end, researchers use F norm with regular term to reconstitute Loss function showing below:

$$L = \sum_{u=1}^{n} \sum_{i=1}^{m_n} \left\{ \frac{w_{u,i}}{2} \cdot \left[\hat{g}_{u,i} - \mu - b_i - q_i^T \times \left(p_u + \frac{\sum_{i=1}^{m_u} q_i}{\sqrt{m_u}}\right)\right]^2 \right\} + \frac{\lambda}{2} \cdot \left(b_i^2 + \|q_i\|^2 + \|p_u\|^2\right)$$

By contrast the performance with various of Recommend Algorithms on real data, The research by calculate implicit feedback of user demonstrate a higher accuracy and recall value on preference prediction than other models. Nonetheless, considering about the complexity of calculation for preference score by the recommend algorithm. It could be a seriously issue for practical application in real life. Moreover, the amount of user operation data would also has an impact on the performance of the Recommendation System. Therefore, if the complexity of calculation would be reduce from representation selection or algorithm optimizing, The research may have a more widely application.

The various of representations for similarity calculation in former research is showing below:

	Paper	Representations for Similarity Calculation	Researchers
1	Personalized internet bar game recommendation based on deep learning[J]	Chronological behavior of users	HEN Yaowang, YAN Wei, YU Dongjin
2	Mobile Game Recommendation using Touch Gestures	Touch gesture data in time-sequential of users	Hao-Tsung Yang, De-Yu Chen, Ya-Xuan Hong, Kuan-Ta Che
3	Research and Practice of Building a Personalized Recommendation System for Mobile Game Platform Based on Big Data Mining[11]	Download history and frequency history of users	You-Hailang, Qian-Feng
4	Personalized Game Recommendation Based on Implicit Feedback	Implicit feedback data of users	YU Dong-jin, CHEN Cong, WU Jian-hua, CHEN Yao-wang

Table 2.1: Former researches base on various of representations of data

2.2 Recommendation Algorithm

According to different classification indices, the recommendation system could apply by different kinds of classification methods, to calculate the similarity between articles. Most recommendation systems could are base on Demographic-Based recommendation, Collaborative Filtering-based Recommendation Collaborative Filtering-based Recommendation, and Hybrid Recommendation.

2.2.1 Demographic-Based recommendation

Demographic-Based recommendation invariably collects the users' profile, which is mega data like gender, age, and active time, etc. The demographic-based recommendation made a primary hypothesis that "one user may have a high preference on the articles which liked by the user who has a similar profile."(Figure2.1) If we try to use Demographic-Based recommendation to make a personal recommendation, the user profile is needed to calculate the similarity with other users. Then picking Top-N users to analyze their preference to ranking recommend candidates.

Thought the calculation of Demographic-Based recommendation is simple, and the calculation process could be quickly done by offline. However, the reliability of recommendation is weak because even Individuals who have entirely the same profile would purchase different articles. That makes the Demographic-Based recommendation could not describe the actual connection between articles. Furthermore, the explanation of Demographic-Based recommendation is base on the same user profile. Nevertheless, the uncertainty of user profiles' similarity could be a problem for demonstrate rationalisation of this explanation.

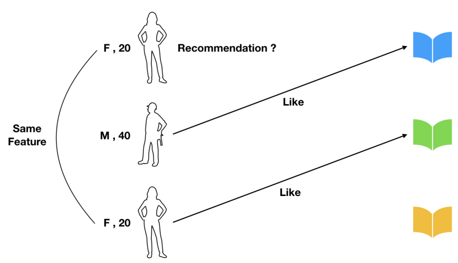

Figure 2.1: Demographic-Based recommendation

2.2.2 Collaborative Filtering-Based Recommendation

The Collaborative Filtering-Based Recommendation refers to the collection of user history behavior, to obtain explicit or implicit users‘ preference information, to calculate the similarity of articles or content, or the similarity of user's profile. Then try to do recommendation base on these relevance(Figure2.2). The collaborative filtering-based recommendation could be base on user, item, or model. Both user-based method and item-based method are belong to the memory-based method, which means base on the history in the system to do the preference prediction. In real calculation, the articles' preference data from users invariably be a large sparse matrix. aim to decreasing the calculation, we usually import clustering method the Collaborative Filtering-Based Recommendation.

Unlike demographic-based recommendation, the user-based method under collaborative filtering-based recommendation calculate the similarity between user by the preference of history, rather than user themselves. Which made the hypothesis that the users bring the similar preference would give similar score on all items.

Item-based method under collaborative filtering is patent algorithm of AMAZON [13]. It has one same hypothesis with content-based method that user could like the article which similar with they like before. different with content-bases method, the "similar article" here is calculate by the preference score before, rather than the item itself. For a application of the number of user larger than the number of item, Item-base method under collaborative filtering-base recommendation could have better performance than other method.

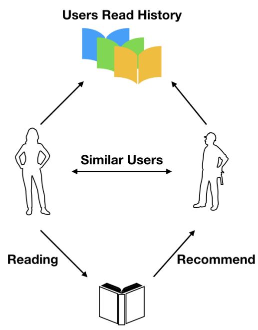

Figure 2.2: Collaborative Filtering-Based Recommendation

2.2.3 Model-Based Recommendation

Given the user-based method and item-based method would face data sparseness issue, making the method hard to process under the big data situation. Model-Based Recommendation was developed, it would train a model by history data, then use the trained model to make preference prediction: Latent Semantic Indexing and Bayesian Networks widely used in the Model-based Recommendation. The input value is the score of representations data from Item Profile and User Profile, while the output value is the preference score of users. Using functions to fit or regress one model, or change the scores to categories for turn into a multi-classifier problem. Nevertheless, the Model-Based Recommendation is usually facing the incremental training problem, which could affect the accuracy of the model.

2.2.4 Hybrid-Base Recommendation

Hybrid-Base Recommendation is another research hotspot of the recommendation system. It mixes multiple recommendation method, aiming for better accuracy(Figure2.3). For solving the Cold-Start problem, it always combine collaborative filtering-based recommendation with other methods. The normal mix method showing below:

Weighted: The weighted method would give weight to the consequence of various of recommendation system. For example, give same weight to the collaborative filtering-based method and the content-based method. Then keep to adjust these weight after calculate the minimum loss value between prediction and users' real preference.

Switch: According to different issue to apply different recommend methods. For example, the system would first try content-based recommendation, and observe its performance. As long the performance is not well as hypothesis, it would automatically change to collaborate filtering-based recommendation method. Even this method would burden the complexity of the system, but this method would be sensitive to the strengths and weaknesses of various of recommended technologies.

Feature Combination: This method would combine the feature from different candidates, and recommend by another recommendation technology. For example, normally we would add the information from collaborative filtering-based recommendation as the new feature vectors. Then apply content-based recommendation method to these data. The feature combination method decrease the sensitivity of item preference from users.

Cascade: The cascade method means using one recommendation method to optimize another recommendation method. First it would use one recommendation method to make a rough candidate results. Then base on this rough candidate results, the cascade method would apply another recommendation technology to do precise recommendation. The cascade method would allow certain the well-distinguished candidates or the candidates which bring few comments remain after the first stage, to avoid these candidates filtering by other recommendation method in another stage.

Feature Augmentation: This method using the output from former recommendation system as the input of next recommendation system. Different with the cascade method, the output from former recommendation in feature augmentation method is not recommendation results, instead, it is certain features which provide for the next recommendation stage. Like using classify labels from clustering method to explore the relevance of various of articles.

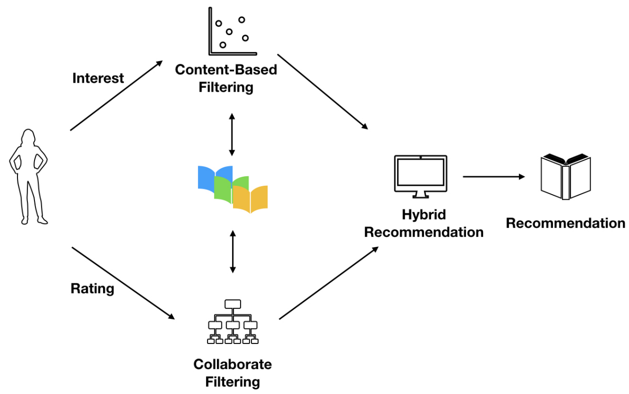

Figure 2.3: Hybrid-Base Recommendation

2.2.5 Content-Based Recommendation

The demographic-based recommendation usually called one description of User Profile. While the content-based recommendation corresponds to another description of User Profile. The content-Based Recommendation has one hypothesis that "one user could like the article which similar to the article he liked before." unlike the Item-Based Recommendation under Collaborative Filtering-Based. "the article he liked before" refers to the profile calculated by the history of users. For example, the simplest way to do content-based recommendation is to calculate the similarity distance, which between recommend articles that user never seen and articles that user liking right now or before. Then, make the recommendation list according to the similarity value. There is no doubt that the content-based recommendation could be complexity in certain times, we could apply decision tree model or neural network model in the recommendation system, but the core of these method is still calculate the similarity distance between User Profile and Item Profile.

Under most Content-Based Recommendation system, the User Profile would not be explicit data. The system usually directly uses the articles liked before or the article user liking right now to list the recommendation candidates. As an example showing in the picture (Figure 2.4), to predict users whether like one new book or not, the recommendation system would calculate the similarity between the book user read before or the book user is reading. A host of experience [12] implied that this method is more flexible also comes with a better consequence.

The Content-Based Recommendation would not face the Cold-Start problem when a new article was add-in. The reason is that as long the Item Profile exists, the similarity between other articles could be calculated. Meanwhile, the explanatory of the Content-Based Recommendation is usually higher than other recommendation systems. Therefore, according to the lateral contrast of most Recommendation system advantage, and considering about the distinguishable visual representations of the games, could be regarded as the content of the articles, this research is base on the Content-Based Recommendation.

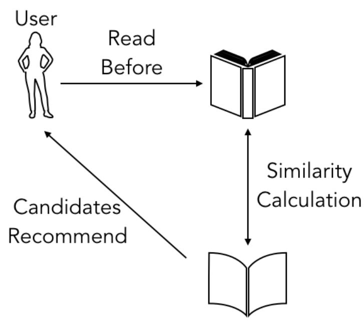

Figure 2.4: Content-Based Recommendation

2.3 Neural Network

2.3.1 Unsupervised Learning in Neural Network

In Machine Learning and Cognitive Science filed, Artificial Neural Network (ANN) is an algorithm and mathematics model which attempt to simulate biological. neural network use massive coupling neurons to do calculation or estimate or approximate one function. Each neurous is an output function (Activation Function), the connection between two neurous is weighted value. There are three unit for process data, the input unit, the output unit, and hidden unit. The input unit mission is receive signal or data, the output unit would output the consequence of neural network processing. The hidden unit exist between input and output units. The connection weight between neurons reflects the strength of the connection between the units. The Neural Network could alter its inner structure once receive the information by input unit from outside. That make the Neural Network an adaptive network which brought learning capability. Neural networks are usually optimized via a learning method based on mathematical statistics[14].

There are two learning trends in Machine Learning field, the supervised learning and unsupervised learning. The training sample in Supervised Learning bring labels, while the training sample without label would be use during training process in unsupervised learning. In real world, given most of data would be collect without label. Therefore, the unsupervised learning is widely use than supervised learning. According to the Calculation principles, some popular supervised learning method like PCA and Isomap could also be used in Deep Learning. However, the complexity cost of these methods is too expensive, also the Secondary information could be miss by these methods. Thus, there are two unsupervised learning methods generally use in deep learning: Autoencoder method and its improvement, aiming to reconstitute original data from abstracted data. Another is Restricted Boltzmann Machine and its improvement methods, aiming to ensure the probability of original data‘s appearance would reach maximum, when Restricted Boltzmann Machine becoming stable.

2.3.2 Autoencoder

Rumelhart first proposed autoencoders in 1986[15]. Under the significant data era, researchers are often facing a dimensional disaster because of vast amounts of resources. The dimensional disaster made the process of data analysis tough to apply. Therefore, this situation driven autoencoder algorithm became a new hotspot in dimensional decreasing of deep learning. The autoencoder algorithm could process high dimensional sparse data efficiently, and usually, it did by the PCA method. It could reduce unrelated and redundancy data via the un-supervised method for dimensional decreasing, according to the characteristics of game image frames data in this research. The upgrade autoencoder would be used for the representations extraction process to extract the core information of image frames in games, and use this information to explore the similarity between games.

Autoencoders algorithm is one typically triple neural network structure. Including input layers, hidden layers, and output layers. Assuming the dimension of input layers and output layers is n, the dimension of hidden layers is m, the sample dataset is $S = \left\{x^{(i)}\right\}_{i=1}^{N}$

Then we assume the function of encoding is f , the process of encoding refers to the process from encoding layers to hidden layers. The assumption of decode function is g , the process of decoding refers to the process of hidden layers to decode layers.

The encoding part of Autoencoders means using nonlinear mapping function to mapping the input data into the hidden unit. Supposing h represents activating of neural network units in the hidden layers. We have h showing below:

$$h = f(x) = s_f(wx + p)$$

w in the function represents the weight matrix between input layers and hidden layers. Meanwhile, s_f means the active function of encode models, which normally use Sigmoid function $(f(x) = 1/\left(1 + e^{-x}\right))$

The calculation principle of decode models is sectional similar to encode models, using the hidden layers from the encoding part to reconstitute the original data. The calculation principle of decode models is showing below:

$$y = g(h) = s_g(\widetilde{w}h + q)$$

The y represents the reconstitution of input data by decode models, s_g represents the activation function of decode models, usually using Sigmoid function or identical function. The $\widetilde{w}$ is the weight matrix between hidden layers and output layers. Also, the parameter θ of autoencoder equal to $\{w, p, q\}$.

The output data y could be regard as the prediction of input data. If the distance between output data y and input data x is acceptable for us. Therefore, the autoencoder remain most information of original data, that we could stop training the model.

By using reconstruction error function $L(x, y)$ to describe the distance between the y and the x :
Assuming s_g is identity function, then we have:

$$L(x, y) = \|x - y\|^2$$

Assuming s_g is Sigmoid function, then we have:

$$L(x, y) = -\sum_{i=1}^{n} \left[x_i \log y_i + (1 - x_i) \log (1 - y_i)\right]$$

When the sample of training dataset is $S = \left\{x^{(i)}\right\}_{i=1}^{N}$, the Loss function of the Autoencoder become:

$$J_{AE}(\theta) = \sum_{x \in S} L(x, g(f(x)))$$

By repeat to calculate iteration of $J_{AE}(\theta)$ for the minimum value, then we could solve the θ parameter of Autoencoder neural network, to complete the training of Autoencoders(Figure 2.5)

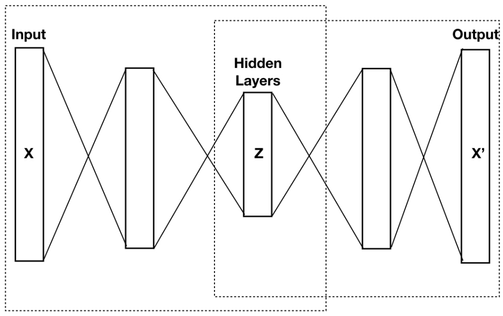

Figure 2.5: Autoencoder

As we apply the upgrade autoencoder in our research to do frames representation process. We expect this technology would adjust its weight to the best state to catch the most basic features from the images after training epoch and turn these basic features into the simplest structure for the description of these primary objects, background, or characters in the frames. Then, we would try to use these descriptions to represent each frame to prepare for the recommendation database building process. Moreover, To predict the user preference, the well-trained autoencoder would extract the representation of the frames which users are playing right now for prepare to calculate the similarity distance between user playing frames with the frames in recommendation database.

Chapter 3

Methodology

In this Chapter, we introduce the methodology about data acquisition and preprocessing, the representations data extraction model, and the recommend database building method used in our research.

3.1 Data Acquisition

3.1.1 Image Resource

In this research, for the similarity calculation part, we would extract the visual representations of games from the image frames data of games. There are some accessible public image data like MINIST, ImageNet, MS-COCO, Open Images dataset, VisualQA, and so on. They are collected and labeled by the professional research teams from Cambridge or Oxford. These data are ideally suited for training in the graphic processing of deep neural networks. Furthermore, in some transfer learning, quite a lot well-trained model or half well-trained model was trained by these data. The researcher would directly apply these models or adjust the last few layers' weight or change the output layer to make the prediction.

Nevertheless, there are not abundant public games that the dataset exists. Most online free game data are about other aspects of games, like tags of games, games history, some scoreboards of famous games, or behavior data of users. As for image frames of games, despite the image frames of games are image data. However, unlike the standard image data, the image frames of the game are hard to obtain. Since the games industry belongs to the cultural industries, notably, different image frames of games may hide the main story line or represent the crucial features of games, which should only appear for the paying players. So when we are talking about image frames or videos of games, the copyright problem is unavoidable. It is, of course, we could see this problem from more and more esports copyright lawsuits recently.

Therefore, in summary, we would use private data which obtain by a Web crawler in this research.

3.1.2 Data Acquisition Tool

The web crawler is one technique that sends the request to the target site and extracts information automatically from the website. Nowadays, in the big data mining era, the web crawler plays an import role in data collection and storage, especially the data resource is not public or arduous to acquire.

The most shared web crawler is a search engine like Google, Yahoo, etc. In this research, data collection applied by the universal web crawler. Furthermore, the data source we selected website STEAM, the largest comprehensive digital distribution platform of the game. It was launch in 2003 and developed by American game company Valve. It also occupied 75% market shares in 2013.

Since the game has some storyline like films, and one game is makeup by different scenes, that makes it arduous to defined the whole game by one picture or one scene. Moreover, that makes the similarity calculation in our recommendation system should base on the various scenes of the games, rather than the whole game. Therefore, though different scenes belong to one game, the thinking of Machine Learning becomes un-supervised learning.

Considering the scenes character of games, we decided to pick the advertising videos of the game as a data resource in-game introduction page. Due to the advertising videos of the game has the most splendid and representational scenes, therefore, assuming as long the advertising videos are available, the visual representations of one game could be extracted.

Given different labels involve various representative games, in this research, we would set the most popular game labels as the data resource for data collection and obtain various of games from each labels to build our recommendation database. The whole working flow of web crawler(Figure 3.1) is showing below:

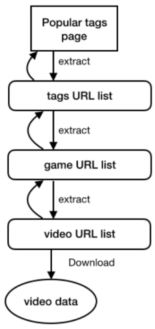

Figure 3.1: Web Cralwer Work Flow

3.2 Data Preprocessing

In Machine Learning or Data Analysis field, the data preprocessing part usually is one of the most vital processes of the whole project. The quality of data would be easily affected by noise data, absence data, and inconsistent data. Since the data we have, most time, is mainly from a different source of different structures, therefore, the unsuccessful data could lead to an unsuccessful Machine Learning or Data Analysis project, what so-called "Trash in Trash out."

There are plenty of factors involve in the quality of data, like accuracy, integrity, consistent, efficiency, reliable, explainable. The fundamental elements inaccuracy, incomplete, and inconsistent are three main influencing factors of most low-quality datasets in the world. Multiple causes could result in inaccuracy data: the web page of information collection has bugs, the server's problem occurs, the mistakes made by the administrator during the input process, repeat application of data, or the wrong code during the data transmission process. The reason for incomplete data, like some privacy data, are unavailable, the wrong comprehension of data form, delete by mistake, ignore of input, or equipment malfunctions could all lead to incomplete data. As for inconsistent data, depending on the different formats of data, or different application rules of data may result in an inconsistent problem. All these uncertainly data could have the effect of performance and output of the model.

Especially in this research, the video data collected by the web crawler.

Without any preprocessing, the data hardly used for training one model. Moreover, the visual representations of games could not be extracted from dirty data. The preprocessing in this research not only decides the similarity of games but also decides the whole recommendation system performance.

The first preprocessing of data in this research is to remove duplicates games. The video data crawled by the web crawler are base on the different tags, considering one game could be categories by different tags like some action games also have strategy operation inside the games.

Therefore, the game could exist in both action and strategy tags, so the first mission is removing these duplicate games. Due to some games that do not have introduction videos, after wiping off 12% duplicate games by contrast function, the blank video of games also been wiped off. As a result, we have 350 games video as the data pool.

Since the recommendation based on the scenes of games. Therefore, we had to make the most significant frames from the video. After watching some of the advertising videos, we found a characteristic that, usually, most of the advertising video would have 10 15 seconds screen prompt text (Figure 3.2) to capture gamers at first. This meaningless scenes should not involve into clips of games, due to their high similarity.

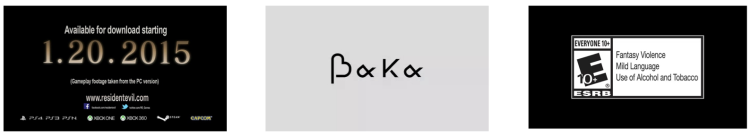

Figure 3.2: Prompt Text of Advertising Video

After did incision of games advertising video, we take the screenshots of the left video clips. Considering the high similarity between advertising videos and films, most of the shots would stay for 4 5 seconds to tell the story for every significant scene. Then we wrote the screenshot function to set interval time and random cut ten image frames from loop playback video clips of each game. As for the part of advertising videos not long enough to produce ten image frames, we adopted some augmentation and rotation to enrich the dataset. About the image normalization process, since the video clips of games have different sizes depend on game upload, all image frames cut into (225,225) uniform format. Moreover, we also face some color channel problem, a few images of frames not based on the RGB channel, then wipe these data off by manual operation.

3.3 CNN-Autoencoder

The most advantage of the autoencoder is the high ability of expression. It could process high dimensional sparse data efficiently. By un-supervised learning, it could reduce unrelated and redundancy data, for decease the dimension. Therefore, this research takes advantage of autoencoder, aim to extract the core information of game image frames to represent the visual representations of the games. Then calculate the similarity of games based on the core visual representations, which extracted by the Autoencoder model.

CNN-Autoencoder is an unsupervised method that combines the traditional autoencoder method with the convolution and pooling method in the Convolutional neural network.

3.3.1 Convolutional Operation

First, look into the convolutional neural network theory. Normally under continuous state, convolution is considered to be a multiplicative integral of two functions (or signal) after reversal and displacement:

$$f(t) * g(t) \int_{\infty}^{\infty} f(\tau)g(t-\tau)d\tau$$

As a result, the convolution operation could produce one new function (or signal). Therefore, the convolution also satisfies the formula:

$$f(t) * g(t) = g(t) * f(t)$$

In 2-dimensional disperse space, the convolution operation could be defined as:

$$O(i,j) = \sum_{u=-\infty}^{\infty} \sum_{u=-\infty}^{\infty} F(u,v) I(i-u, j-v)$$

Given the range of images is limited, the formula could be written into:

$$O(i,j) = \sum_{u=-k-1}^{k+1} \sum_{u=-k-1}^{k+1} F(u,v) I(i-u, j-v)$$

$\cdot O(i,j)$ represent the output pixel, located in (i,j)
$\cdot 2k+1$ means one edge of rectangle odd convolution kernel
$\cdot F$ is convolution kernel
$\cdot I$ is input image

The filter operation would do smoothing operation to input signal, for extract partial information from it(Figure 3.3).

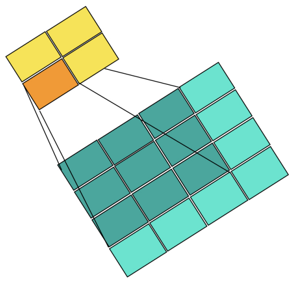

Figure 3.3: Convolution Operation

The result of filter operation depends on the value of the convolution of the kernel. According to various kernel settings, every convolutional kernel could be used for different processes of the image, like noise reduction or fuzzy processing.

There are also two additional parameters of 2-dimensional disperse convolution operation, horizontal and vertical move steps. They are the jump pixel along each dimension of image I, after done the operation convolutional operation. the two parameter usually has the same value, and they are equal to S.

For a square image i.e.$IW = IH$. After 2-dimensional disperse convolution operation by step $2k+1$, we could get a image O:

$$O_w = O_h = \frac{I_w - (2k+1)}{S} + 1$$

If the image has multi-channels, then the convolutional operator should be operated on each channel. Moreover, the number of output channels in one convolutional kernel should invariably be the same with the number of input channels. In conclusion, a 2-dimensional disperse convolution operation is a stackable process to the signal.

About the convolution of each dimension. Assuming the cuboid could be represent by a triple tuple (W, H, D). It is obviously that grayscale image could be see as a cuboid which depth D equal to 1. Meanwhile, the RBG image could be see as a cuboid which depth D equal to three. A convolutional kernel consider to bring depth D too. Therefore, we could assuming the image and filter are gather of single channel image and single channel filter.

$$I = \{I_1 ... I_D\}, F = \{F_1 ... F_D\}$$

Considering the depth of image, then the convolutional formula could be change as:

$$O(i, j) = \sum_{d=1}^{D} \sum_{u=-2k-1}^{2k+1} \sum_{u=-2k-1}^{2k+1} F_d(u, v) I_d(i - u, j - v)$$

3.3.2 CNN-Autoencoder Model

Unlike the normal convolutional, the filter of Convolutional Autoencoder (CAE) defined the mission from the different aspects. Once these standard convolutional filter has trained, they would be applied to do visual representations extraction from any input data. Then use these visual representations for the various mission like classification.

CNN-Autoencoder is involved in Convolution Neural Network. However, Convolution Neural Network is supervise learning, training for core representations extraction to do classification. On the contrary, Convolutional Autoencoder is unsupervised learning, used to do reconstitution input data after visual representations extraction in this research.

encoder

In this research, Every convolutional layer is made-up by n (hyper-parameter) of convolutional kernels. each depth of convolutional kernel is D, which represent the channels of input data.

Therefore, use one group of convolutional kernel:

$$I = \left\{F_1^{(1)} ... F_n^{(1)}\right\},$$

to do convolutional operation on each image with depth D:

$$I = \{I_1 ... I_D\}$$

As a result, produce n of the equivalent feature graph or active graph:

$$O_m(i,j) = a\left(\sum_{d=1}^{D}\sum_{u=-2k-1}^{2k+1}\sum_{u=-2k-1}^{2k+1} F_{m_d}^{(1)}(u,v)I_d(i-u,j-v)\right) m = 1,\ldots,n$$

Aim to improve the generalization ability of the neural network. Each convolutional kernel would be active by a nonlinear function. By this training method, the neural network could also learn some nonlinear features of input data.

$$z_m = O_m = a\left(I * F_m^{(1)} + b_m^{(1)}\right) m = 1,\ldots,m$$

b_m^i is bias value of feature graph m . the active graph is recode of input data I, make it could be demonstrate in lower dimensional space. The dimension is no longer equal to dimension of original O after reconstitution. However, these parameters are learning from O_m, in another word, these parameters are need to be learn in CNN-Autoencoder.

Since the target is reconstitution from the feature graph production of input data I. Therefore, the decode operation is need. The convolution operation could be do again in decode, because CNN-Autoencoder is a completely convolutional neural network.

For the reason that convolutional operation would decrease the range of output. Then, we could not use the convolution to reconstitution information with the same range of input. However, the input fill operation could use to tackle the problem. If the fulfill by zero into input data I, after the first convolution, the result would have more range of space than input data I . As a result, it could produce the same range of space with input data I after the second convolution:

$$dim(I) = dim(decode(encode(I)))$$

That means we would fulfill $2(2k+1)-2$ of zero to data I. (each edge with $2(2k+1)-1$ of zero. That make the width and height of convolution encode to:

$$O_w = O_h = (I_w + 2(2k+1) - 2) - (2k+1) + 1 = I_w + (2k+1) - 1$$

In this research, the whole model is using Machine Learning structure Keras model API.he three convolution layers of the encoder built by Keras function Convolution2D. There are 32 output of Convolutional kernel of the first and second layer, each size of them is 3*3, and 64 output of convolutional kernel of the third layer, all these filters F are result of convolution operation. The nonlinear activation function a was set as Relu. Also, aiming to decrease the calculation and parameters, three of MaxPooling layers also added in the encoder.

decoder

For reconstitution of image I, the compressed information n of feature graph $(zm = 1, ...n)$ is use as input for decoder.

Given convolution operation would be done on every feature graphs, and produce $(2k+1, 2k+1, n)$ of dimensions. So, after convolution by filter $F_{(2)}$, It would produce the same range of I.

The number of filters need to be learned is D, since we have to reconstitute the input image with depth D.

Therefore, the reconstitution image $\widetilde{I}$ is a production of convolutional operation, by the feature graphs' dimensions $Z = \{zi = 1\}^n$ with its convolutional filter $F_{(2)}$.

$$\widetilde{I} = a\left(Z * F_m^{(2)} + b^{(2)}\right)$$

According to the above-mentioned zero fulfill operation, the dimension of decode after convolutional operation becoming:

$$O_w = O_h = (I_w + (2k+1) - 1) - (2k+1) + 1 = I_w = I_h$$

There are four convolutional layers of decoder build by Keras function Convolution2D in this research. The 64 output of the convolutional kernel of the first layer's size is 3*3. Each of the second and third layers has 32 output of convolutional kernels — all size set to be 3*3. The first three layers of decoder would bring a nonlinear activation function of Relu the same as the encoder.

As for the decoder structure, it also brings three MaxPooling layers. There are three up-sampling layers for reconstitute original images. The final layer has 3 outputs with a size 1*1 filter. The CNN-Autoencoder would first receive the frames with standard size and put into the encoder, for extracting the essential image representation from background or objects in the frames. After the model gets the basic description of the frames, it would attempt to reconstitute the original data based on the basic description of the frames. When the whole flow finishes, aiming to evaluate the training and do the preparation of the next epoch training, the CNN-Autoencoder would calculate the loss value between original data and reconstitute data. Then, adjust the weight of the structure and complete the self-learning process. The whole CNN-Autoencoder structure is showing below(Figure 3.4).

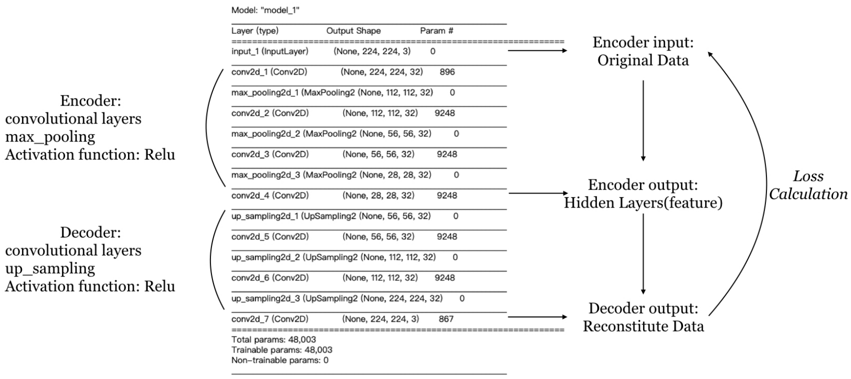

Figure 3.4: Autoencoder Structure in this Research

In this research, the model above would train on the frames of the games dataset. After the training process finished, aim for visual representations extraction, the trained model would be used on both frames of games dataset and users' image(Figure 3.5).

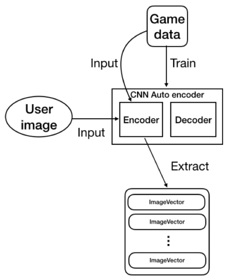

Figure 3.5: Autoencoder Workflow in this Research

3.4 Clustering

In this research, In order to speed up the similarity calculation of user image with games frames data in the recommendation system. We would clustering all games frames into K of clusters to build recommendation candidates database in advance, each cluster is K_n. When the recommendation system receive the frames F which user are playing in real time, the recommendation system would calculate distance between F and the centroids of cluster K_n. As the closest cluster determined, the recommendation system would calculate the distance between F and members frames of cluster K_n, furthermore, produce the recommendation list from Top-N closest frames of games.

Assuming we have 1000 frames database with one user frames, The clustering method reduce the complexity of calculation from $1 \times 1000 = 1000$ times to $10 + 100 = 110$ times. Therefore, given the efficiency with the clustering method, we would apply clustering method in this research to speed up the recommendation process also reduce the system burden.

3.4.1 KMeans Clustering

The Kmeans Clustering is an unsupervised learning, as this research, the frames of games have no label y but only bring visual representations data x. The target of cluster operation aims to find the hidden y of each sample, and put all samples together with same hidden y.

In clustering problem, assuming we have training sample $\{I^1...I^m\}$, each $x^{(i)} \in R^{(n)}$, The algorithm is showing below:
· Random pick of cluster centroids $(\mu_1, \mu_2...\mu_k \in R^{(n)})$
· Repeat until convergence:

For each sample point i, calculate which hidden y it belongs to.

$$C^{(i)} = jargmin \left\| x^{(i)} - \mu_j \right\|^2$$

For each hidden y, recalculate its cluster centroid.

$$\mu_j := \frac{\sum_{i=1}^{m} 1\left\{c^{(i)}=j\right\} x^{(i)}}{\sum_{i=1}^{m} 1\left\{c^{(i)}=j\right\}}$$

k is the number of clusters which defined by user, $C^{(i)}$ represents the cluster which has the shortest distance between sample i. The cluster centroid μ_j is the conjecture of the cluster center point under same hidden y.

The convergence of Kmeans is distortion function :

$$J(c,\mu) = \sum_{i=1}^{m} \left\| x^{(i)} - \mu_{c^{(i)}} \right\|^2$$

J function represent the sum of the squares of distance between each sample point and centroid. Kmeans method is try to take minimum value of J function.

In this research, the training sample of $I = \left\{ x^{(1)} ... x^{(m)} \right\}$ is image visual representations vectors which extract by CNN-Autoencoder encoder model. As for the number of cluster (k), we would use the Elbow Method to determine who multiple clusters of games dataset.

The K-means Method has no evidence for determine the number of clusters automatically. Given the target function of the Kmeans method is to minimize the loss between the sample and the centroid. The squares of loss between the sample point and the centroid are called distortion. Therefore, for one cluster, as the closer of members in the cluster, the distortion would be becoming lower. On the contrary, the loose of members in the cluster, the distortion would grow. According to the deduce, there exists one critical point that makes an extremely distortion improvement; after this critical point, the distortion would decrease slowly. As a result, in this research, we would apply the Elbow Method to observe this critical point of K-Means clustering and determine this critical point, moreover, apply the number to k (the number of clusters).

In the machine learning field, there are different distance methods for calculation, like euclidean distance, manhattan distance, or cosine distance. Since the representations data would typically be calculated by vector form. Therefore, when analyzing the similarity of two vectors, it invariably usually measures the distance by Cosine Distance. Compare to other distance, The Cosine Distance demonstrates a difference in direction, even under the high dimensional space, the Cosine Distance could invariably maintain the property of "same equal to 1, orthogonality equal to 0, opposite equal to-1". Furthermore, according to the dimension affection, the Euclidean distance would have an uncertain range, also the definition becoming obscure. the definition of Cosine similarity is:

$$\cos(A,B) = \frac{A*B}{\|A\|_2\|B\|_2}$$

The Cosine Distance is defined by:

$$dist(A,B) = 1 - \cos(A,B) = \frac{\|A\|_2\|B\|_2 - A*B}{\|A\|_2\|B\|_2} = 1 - A*B$$

In this research, considering the visual representationss of frames of the image is all present by vectors. Under the Kmeans clustering process, the distance between different vectors would be measure using Cosine Distance.

Chapter 4

Experimentation

In this chapter, we present the train test ratio of data setting, the data pre-processing setting, and the CNN-Autoencoder setting in this research. Also, we demonstrate the detail of model training and result of the experimentation.

Here is a slice of sample of data is showing below:

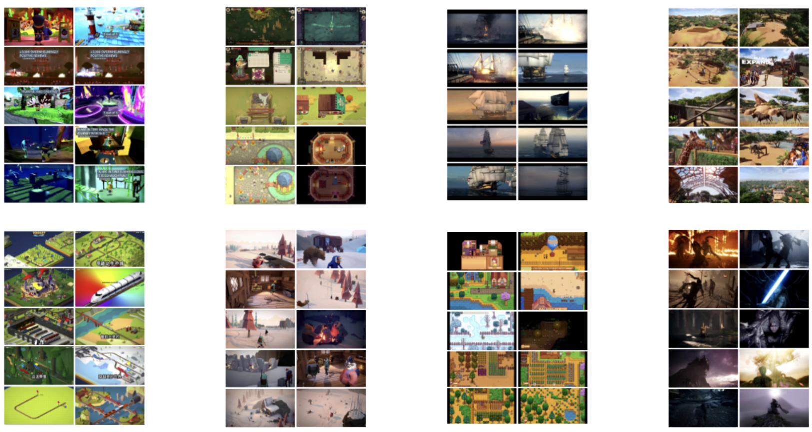

Figure 4.1: Sample of Data

4.1 Test Train split setting

According to train test ratio, we set all dataset split into 7:3, which is frequently-used ratio in machine learning field. In this research, 70% of frames of data would be use for training. While 30% of frames of data would be use for test the whole recommendation system.(Figure 4.1)(Figure 4.2).

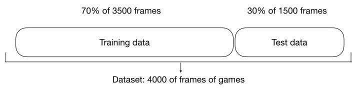

Figure 4.2: Train Test dataset Split

4.2 Data Preprocessing Setting

For satisfying the need for deep learning model input, the image frames data should be converted into an array form. We use KERAS preprocessing model to process the image frames data for training preparation. After analyzing all image frames into an array form matrix, We turn all array data from int form into float form, also divide data by 255 to ensure the data normalize and standardize(Figure 4.3).

Figure 4.3: Data Normalization

4.3 CNN-Autoencoder Setting

The CNN-Autoencoder model is base on the Keras back end. Model training epoch was set to be 50 times, for seek a well training. The batch size was set to be 32. Also we use Adam algorithm as our optimizer.

For every epoch, we will randomly shuffle the dataset, and make a batch to train the CNN- Autoencoder model.

Since the dataset which base on the frames of image is too large for normal CPU to run. Then the parallel computer system is used for CNN-Autoencoder training and visual representations extraction and clustering in this research:

NVIDIA Corporation GP100GL [Tesla P100 PCIe 16GB] (rev a1)

4.4 Clustering Method Import

The K-Means method efficiently decrease the complexity of calculation of recommendation candidates finding process. In this case, without clustering method add in, the calculation complexity would reach 4000 times. While with clustering method, the calucation complexity decrease to 557 times.(Figure 4.4):

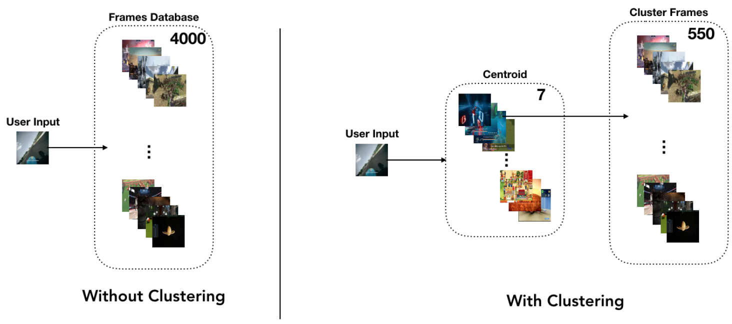

Figure 4.4: The complexity of clustering method import comparison

4.5 Experiment Result

As the training input of frames of games in the training process, the CNN-Autoencoder extract the visual representations of images. Each output of CNN-Autoencoder layers is showing below(Figure 4.5):

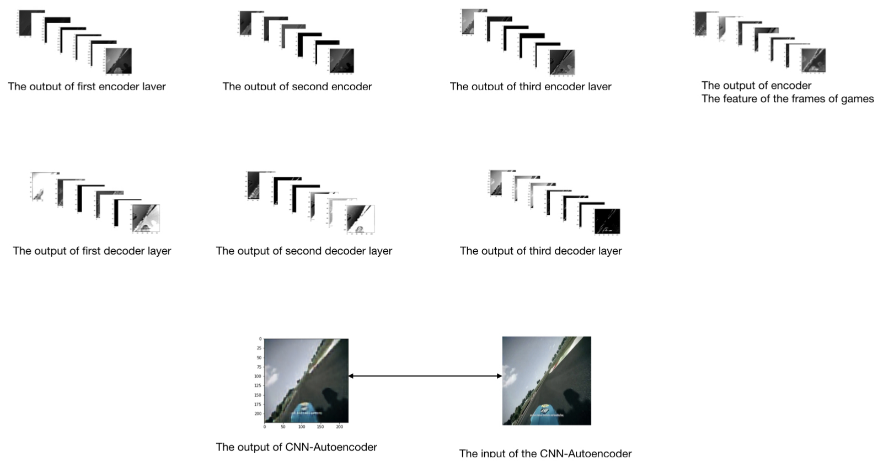

Figure 4.5: Output Sample of Autoencoder

The whole CNN-Autoencoder training process took over 3 hours and the accuracy reach 0.25 value.
In this research, the Elbow Method calculate the reasonable number of clustering is 7.0. The Elbow Method chart is showing below(Figure 4.6):

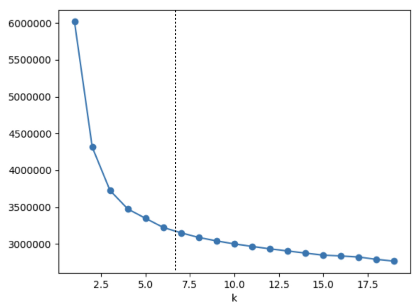

Figure 4.6: Result of Elbow Method

After the whole recommendation system completed. Using the test dataset to testify the performance of the recommendation system. Here is one example showing below(Figure 4.6):

Figure 4.7: Recommendation System in this research

Chapter 5

Evaluation

5.1 Stability Test

The stability of prediction and recommendation would influence the trust to one Recommendation System. So if the recommendation list does not change in a short time or under similar user profile, then, we could conclude the Recommendation System bring a high stability.

Adomavicius and Zhang prompted one algorithm Mean Absolute Shift(MAS)[16] to evaluate the stability of the quality metrics of Recommendation System. They have an hypothesis that, the stability performance of one Recommendation System, should calculate under the variety of user profile enhancement by time change. The MAS formula showing below:

$$stability = MAS = \frac{1}{|P_2|} \sum_{(u,i) \in P_2} |P_2(u,i) - P_1(u,i)|$$

In this research, the shift value would be measure between different but high similarity users inputs. Stability Test will be contrast with three pairs of similar high frames of games randomly. Each output list from recommendation by first user input would Based on their Mean Absolute Shift(MAS) value to evaluate the Recommendation System.

After Recommend Process by received random high similarity users inputs, The consequences of three pairs of recommendation lists is showing below:

Table 5.1: Random Pairs contrast

	Game1	Game2	Game3	Game4	Game5	Game6
1	$40G56s$	$40G56s$	$285G32s$	$285G32s$	$327G22s$	$327G22s$
2	$142G33s$	$236G72s$	$232G8s$	$232G8s$	$68G33s$	$68G33s$
3	$30G22s$	$142G33s$	$223G17s$	$223G17s$	$68G9s$	$149G25s$
4	$102G12s$	$82G17s$	$168G67s$	$168G67s$	$149G25s$	$68G9s$
5	$82G17s$	$30G22s$	$60G33s$	$60G33s$	$277G46s$	$277G46s$
6	$236G72s$	$142G6s$	$216G8s$	$216G8s$	$68G17s$	$68G17s$
7	$82G14s$	$200G72s$	$281G9s$	$281G9s$	$172G14s$	$172G14s$
8	$46G8s$	$82G14s$	$216G9s$	$216G9s$	$81G51s4$	$258G6s$
9	$200G72s$	$102G12s$	$321G17s$	$120G13s$	$258G6s$	$81G51s$
10	$142G6s$	$99G8s$	$120G13s$	$321G17s$	$114G22s$	$218G29s$

.

By calculation of MAS stability value of three pairs of recommendation lists, we have the value of **3.6**, **0.2**, **0.444**. The average value of MAS stability value is **1.414**. Since the value fluctuate between 0 to 5 under 10 pairs of 3 groups, then We could observe that, except the first pairs of the contrast has a slightly shift value on the recommend list order, the whole Recommendation System could give a stable recommend list between high similarity of users input.

5.2 Case Explanation

In this research, given the data set took the frames and scenes from games as a unit, instead of the conventional way of whole games as a unit for various games' similarity calculation. Various representative frames exist in one game, and different frames in one game might have an exceptionally long distance between each other in similarity calculation. As a result, the unsupervised learning idea applies to the Recommendation System for making the recommend list in this research. On the other hand, it causes the accuracy of the Recommendation System is an inconvenience for the measure. In summary, to evaluate the Research Consequence, We would process various test data into the Recommendation System and observe its describing performance of the main visual representations in frames.

The evaluation data would split into three groups. Each group would randomly select 10 of the frames of games from test data as input into the Recommendation System one by one. After all recommendation lists have made, we would observe the similarity of each case between input frames and candidates frames in the recommendation list. The precisely of the Recommendation System would be measure from describing of how many the representative objectives both frames bring.

In cases explanation, we would pick four representative cases to give explanations of the performance of the Recommendation System in this research.

Case1:

The playing game of user in real time and recommendation list of Case1 is showing below:

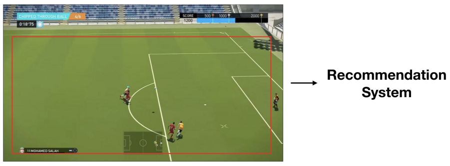

Figure 5.1: Case1 user input

Figure 5.2: Case1 Recommendation List

The description of Case1: When User input (iteration: 1,frame: 2) being a soccer game for testifying. As the experiment consequence demonstrate that, the recommendation system precisely catches the grassland object (Red Rectangular in user input(Figure 5.1)). The output of the recommendation system is to share a similar object(Red Rectangular in Recommendation candidates(Figure 5.2)) of green grassland. Moreover, There are three frames from one game [*Football New Star Manager*] bring the same grass and football objects.

Case2:

The playing game of user in real time and recommendation list of Case2 is showing below:

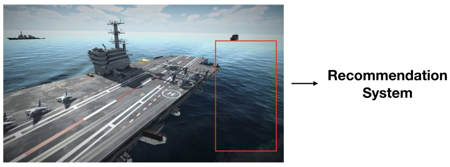

Figure 5.3: Case2 user input

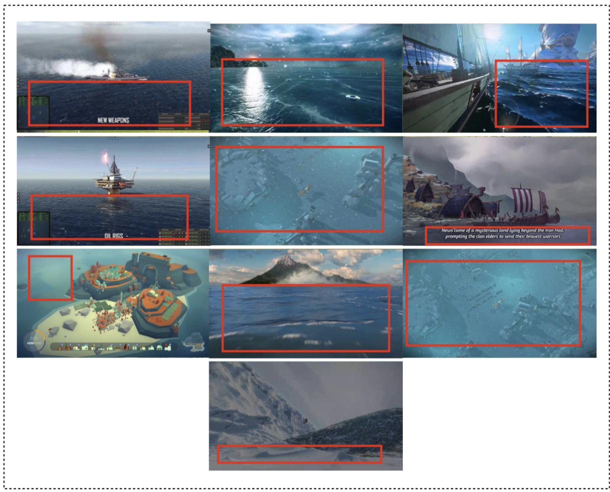

Figure 5.4: Case2 Recommendation List

Case2: When User input (iteration: 2,frame: 2) being a sea battle game for testifying. The recommendation system precisely catches the ocean object(Red Rectangular in user input(Figure 5.3)). The output of the recommendation system shares the similar object(Red Rectangular in Recommendation candidates(Figure 5.4)) of vast ocean or ships. Also three frames from same game [*Cold Water*] show the similar pattern of warship sailing in the ocean.

Case3:

The playing game of user in real time and recommendation list of Case3 is showing below:

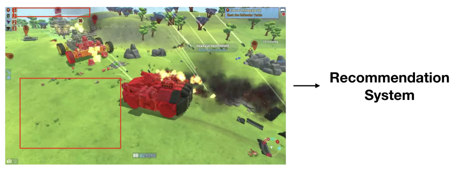

Figure 5.5: Case3 user input

Figure 5.6: Case3 Recommendation List

Case3: When User input (iteration: 3,frame: 8) being a battle game on lawn for testifying. The recommendation system catches double-objects(Red Rectangular in user input(Figure 5.5)); some of the output of the recommendation system is sharing the similar object(Red Rectangular in Recommendation candidates(Figure 5.6)) of grassland like frame2, frame4, frame5, frame7, and the last frame. Meanwhile, some are sharing the sky elements like frame6 and frame9. Two of frames from one game [*AutoNuts*] building creative structure on the grassland, while another two of frames from the game [*CONDUCT DELUXE*] which building railway in the countryside.

Case4:

The playing game of user in real time and recommendation list of Case4 is showing below:

Figure 5.7: Case4 user input

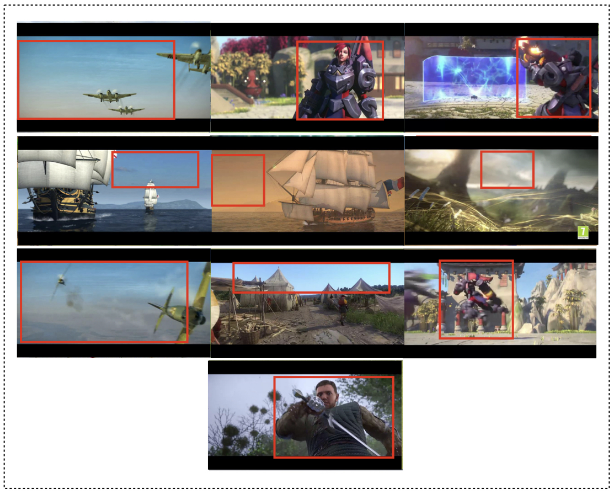

Figure 5.8: Case4 Recommendation List

Case4: When User input (iteration: 2,frame: 5) being a character battle game for testifying. The recommendation system catches multi-objects(Red Rectangular in user input(Figure 5.7)); some of the output of the recommendation system is sharing the similar object of sky flying gesture(Red Rectangular in Recommendation candidates(Figure 5.8)) like frame1, frame4, frame5 and frame7. Meanwhile, some are sharing the characters shape (Red Rectangular in Recommendation candidates(Figure 5.8)) like frame2, frame3, frame9 and frame10. There are two games have the maximum of occurrences in the recommend list. First game [*IL2 Great Battle*] has two frames show the air fighter battle. The second game [*PALADINS: Champions of the realm*] has two frames bring the same character outlook shape.

Chapter 6

Conclusion

There are both favorable and unfavorable performance exist in this research.

In Case1 and Case2, the recommendation system perfectly catches the main objects of the scene, also give all suggestion with the objects in the frames. In Case4, the recommendation system catches multiple objects and hard to calculate the particular scene to make the recommendation. Moreover, we could observe as higher the weight of specific objects in the original frames of Users' playing games. The same objects would also have a higher occurrence in the frames of the recommended list. In this research, after analysis games from recommendation list.

We also certify the inaccuracy of the Recommendation System, which base on the tags of the games, like in Case1, the tags of user input game bring the tag of football and action. According to the graphic similarity calculation of Recommendation System in this research, the game has the closest distance with the user input game is *Football New Star Manager*. However, the *Football New Star Manager* bring tags of strategy and sports management, which could cause a high deviation in the tags-base Recommendation System.

However, various types of behaviors of individuals exist on the internet. It is typical for the user that abruptly try something new, or click the recommend article with no reason. Therefore, concerning about this uncertainty, it is uneasy to predicting the preferences of these users, and evaluate the performance of certain Recommendation Systems in real-life.

Future

We should admit that the research still needs more research to focus on the graphic processing section. The whole research could be extracted with more games to support the recommendation system database and give more variety of recommendations.

Also if the recommendation system could obtain more frames of user playing and try to weighted average representation of users' preference. Then the system might be able to distinguishing the weights among multiple objects, and focus on the main objects which brought the highest weight to make the recommendation, the accuracy of the Recommendation System might also could be development.

Bibliography

[1] Cai. Research on the investment trend and Prospect of Chinese electronic competitive industry, pages 2-5 (2016)

[2] Ilija Ilievski,Sujoy Roy. Personalized News Recommendation based On Implicit Feedback, *NRS*, October 13 2013, Kowloon, Hong Kong 2013

[3] David Goldberg, David Nichols, Brian Oki, and Douglas Terry. Using Collaborative Filtering to Weave an Information Tapestry, CSL-92-10 September 1992 [P92-00156] Xerox Corporation, Palo Alto Research Center, 3333 Coyote Hill Road, Palo Alto, California 94304

[4] Paul Covington, Jay Adams, Emre Sargin, Google Mountain View, CA. Deep Neural Networks for YouTube Recommendations, ACM ISBN 978-1-4503-4035-9/16/09, RecSys '16 September 15-19, 2016, Boston , MA, USA

[5] Dayi Lin1 · Cor-Paul Bezemer1 · Ahmed E. Hassan. An empirical study of early access games on the Steam platform, *Springer Science+Business Media*, Queen's University, Kingston, ON, Canada,LLC 2017

[6] Andy Donovan, Hyerim Cho, Chris Magnifico, Jin Ha Lee. Pretty as a pixel: issues and challenges in developing a controlled vocabulary for video game visual styles. *JCDL '13 2013*, DOI:10.1145/2467696.2467747

[7] Zhimin,Wang. The Revolutionary significance of file to the development of human civilization *Hundred Schools in Arts 2020,4* page 33 1003-9104(2012)04-0074-06, Beijing Film Academy,2013

[8] CHEN Yaowang, YAN Wei, YU Dongjin, et al Personalized internet bar game recommendation based on deep learning J *Computer Engineering* 2019 45(1) :206209 216

[9] Hao-Tsung Yang, De-Yu Chen, Ya-Xuan Hong, and Kuan-Ta Chen. Mobile Game Recommendation using Touch Gestures, Institute of Information Science, Academia Sinica, 978-1-4799-2961-0/13/©2013 IEEE

[10] YU Dong-jin1 CHEN Cong1 WU Jian-hua2 CHEN Yao-wang. Personalized Game Recommendation Based on Implicit Feedback. Telecommunications Science, *ACTA ELECTRONICA SINICA*Vol. 46 No. 11 (2018)

[11] Hailang You,Feng Qian,Xiangwei Huang,Liangliang Hu. Research and Practice of Building a Personalized Recommendation System for Mobile Game Platform Based on Big Data Mining[J]. Telecommunications Science, 2014, 30(10): 27-32.

[12] WANG Guoxia, LIU Heping. Survey of personalized recommendation system. Computer Engineering and Applications201248766-76.

[13] B. Sarwar, G. Karypis, J. Konstan, J. Riedl, Item-Based Collaborative Filtering Recommendation Algorithms, Proceeding of the 10th international conference on World Wide Web, WWW 2001.

[14] YIN Rui-Gang,WEI Shuai,LI Han,YU Hong.Introduction of Unsupervised Learning Methods in Deep Learning.COMPUTER SYSTEMS APPLICATIONS,2016,25(8):1-7

[15] Rumelhart DE, Hinton GE, Williams RJ. Learning representations by back-propagating errors. Nature, 1986,323(9): 533–536

[16] Adomavicius, G. and Zhang, J. 2012. Stability of recommendation algorithms. ACM Trans. Inf. Syst. 30, 4,Article 23 (November 2012), 31 pages. DOI = 10.1145/2382438.2382442 http://doi.acm.org/10.1145/2382438.2382442