Song Recommendation Using Weight-Directed Graph and Reinforcement Learning

Aulia Adila and 13519100[1]
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
[1]13519100@std.stei.itb.ac.id

***Abstract*— Recommendation system has become one of the important features in some web applications. It could gain user engagement significantly, by offering relevant products or services based on the user's preference. One approach in building a recommendation system is by using the weight-directed graph, which has items as nodes and weighted edges. Items are described as songs which have several attributes related, meanwhile the edge weight is initialized from Natural Language Processing algorithms, then being updated with user's action that will be calculated with Reinforcement Learning algorithms. This paper will examine the recommendation system applied in songs, as this technology has been used in many digital music services which could allow users to access massive numbers of songs.**

***Keywords*—Recommendation system, Weight-directed graph, Natural language processing, Reinforcement learning.**

I. Introduction

Business couldn't grow without customers. The product's profitability is raised through the customer's usage towards those products. Customer's willing to buy or to consume a product or service is depended on the engagement towards the product. User engagement measures whether users find value in a product which can be measured through varies of activities such as downloads, clicks, shares, and more. Those engagements could be increased as the excitements and willingness from user are mounted, by some profitable activities such as purchases, signups, subscriptions, and more. User engagement is highly correlated with overall profitability. That's why, companies in business are competing to gain customer's attentions and preserves the bond of interaction to keep their hearts towards a product or service.

In order to increase customer interaction, companies must evaluate the behavior of their customers to make improvements and enhance the experience. One of the strategy is gaining the users' experience by educating users. The attention of users could move over time. In order to satisfy the need of the customer, the business must provide a service to suggest new things and desires to the user. This suggestion is implemented into a scheme known as the recommendation system.

Basically, a recommender or recommendation system is an information filtering system that attempts to predict an item's rank or choice that a person is likely to send. Many fields of business has used this system in order to increase and improve customer engagement. For instance, websites such as Netflix and Spotify that recommend movie titles and songs/playlists depending on the tastes of the customer, or Amazon that recommends things that are like what you're looking for. The fundamental step of the Recommender System is to extract as many valuable candidates before reviewing and submitting them to users. One approach of method to build the enrich database is by using a graph which harmonizes data from a variety of sources to provide a detailed description of the query entity. This graph is an weigh-directed graph which has taken the basic concept of Graph Theory.

II. Graph Theory

Graph is used for discrete objects representation in 'nodes' or 'points' with its relations between connected objects. In mathematics, graphs are defined as ordered pairs, with two parts: vertices/nodes - edges. A graph G can be notated as G = (V,E) where V is known for non-empty set of vertices or nodes and E set of edges that connected a pair of vertices that could be empty. V is notated as $V = \{v_1,v_2,v_3,..\}$ where vi is represented a vertice with i index. While E is notated as $E = \{e_1,e_2,e_3,..\}$ where e_i is represented a edge with i index.

Relation between an edge and a pair of vertices is notated as $e = (v_i, v_y)$ where e is an edge connecting vertices with index i and index j.

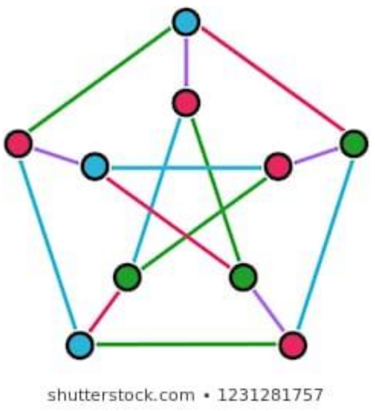

Fig 2.1 Graph Illustration
(Taken from shutterstock.com)

The possibility of generating varies of graph has made graph classified into several types, which will be described below.

A. Simple and Unsimple Graph

Graph could be grouped based on the existence of multiple edges or loop.

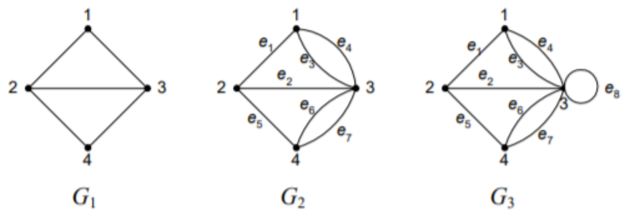

Fig 2.2. a) graph with no multiple edges and loop b) graph with multiple edges c) graph with multiple edges and loop (Taken from Rinaldi Munir Webpage)

Pair of edges are called multiple edges if it connects two same vertices or nodes. As shown in fig 1, G2 has e3 = (1,3) and e4 = (1,3) as edges that connects the same two nodes, which are node 1 and node 3. Thus, e3 and e4 are multiple edges. Meanwhile, an edge is called a loop if it begins and ends in the same node or vertices. As shown in fig 1, G3 has e8 = (3,3) as an edge that connects the exact same node, which is node 3. Thus, e8 is a loop.

Based on the existence of multiple edges nor a loop, graph is categorized into two classifications.

1. Simple graph, which has any multiple edges nor a loop. It is visualized in the picture below.

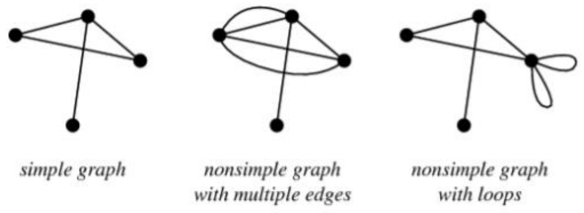
simple graph
nonsimple graph with multiple edges
nonsimple graph with loops

Fig 2.3 Simple graph and unsimple graph (Taken from Wolfram)

2. Unsimple-graph, which is the opposite definition from simple graph. It has atleast one multiple edge or loop. This type of graph is furtherly divided into 2 groups.
   a. Multi-graph, which has multiple edges
   b. Pseudo-graph, which has one or more loops on the nodes

B. Weight-Directed Graph

Weight-directed graph is composed by weighted graph and directed graph. A weighted graph is a graph in which a number (the weight) is assigned to each edge. Such weights might represent for example costs, lengths or capacities, depending on the problem at hand. While directed graph has the type of edges with arrow that connects nodes. It has directional orientation on its edges. Another type of graph with no directional orientation on its edges is categorizes as undirected graph.

Figures below shown the illustration of weighted graph and directed graph.

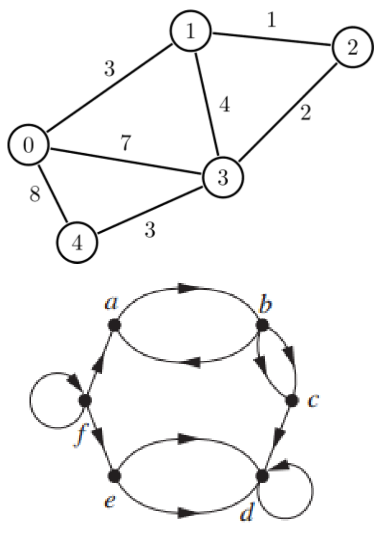

Fig 2.4 Weighted graph and directed graph (Taken from TutorialsPoint and Math Stack Exchange)

In conclusion, weight-directed graph are simple directed graph with weights assigned to their edges/arrows.

C. Terminologies of Graph

Adjacent

Two nodes are called adjacent if they are connected directly through an edge. Based on the figure below, it has known that node 1 has adjacent with node 2 and node 3, while has no adjacent with node 4.

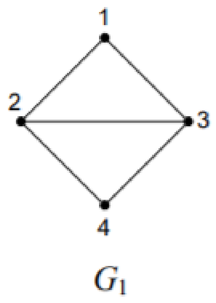

Fig 2.5 Graph with adjacent nodes (Taken from Rinaldi Munir Webpage)

Incidency

An edge and a node that is connected directly are called incidents. For any edge notated by e = (v_j, v_k) there are node Vj and node Vk which have incidence with the edge e. Based on figure G1 below, it has known that edge(2,3) is incident with node 2 and node 3, edge(2,4) is incident with node 2 and node 4, while edge(1,2) has no incidence with node 4.

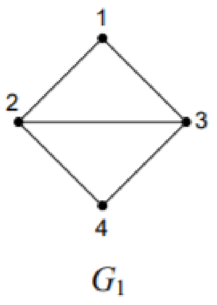

Fig 2.6 Graph with incidents
(Taken from Rinaldi Munir Webpage)

Empty graph

The graph with only vertices and no edges is called null graph or empty graph.

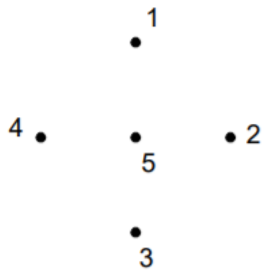

Fig 2.7 Null graph
(Taken from Rinaldi Munir Webpage)

Degree

Degree is a number that represented the amount of edges incidence with a particular node, which notated as d(v). As the figure shown below, for G1, the degrees for node 1 and node 4 has the same value because it has same amount of edges incidence. Thus, d(1) = d(4) = 2

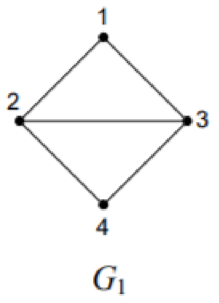

Fig 2.8 Graph Illustration
(Taken from Rinaldi Munir Webpage)

There is a theorem called Handshake Lemma which states that the total number of degrees of graph is twice the number of edges. According to this theorem, we can conclude that a graph has always have an even total number of degrees.

Path

A path in a graph is a finite or infinite sequence of edges which joins a sequence of vertices. It has the starting node v_0 and the final node v_n with the long path which is counted by the number of edges in the path.

Cycle

A path that has the same starting and finish node is called a cycle. It has a length which is counted by the number of edges in the cycle.

III. Natural Language Processing

Natural Language Processing, as usually shortened as NLP is subfield of artificial intelligence that concerned with the relationships between computers and humans using natural language or human language. Computers are trained to handle and interpret large amounts of natural language data. The goal is the system can understand the content of texts, including the qualitative complexities of the vocabulary within them. This technology can accurately extract information and insights embedded in the documents, alongside with organizing and categorizing the document themselves.

Natural Language Processing has the main role as a powerful engine to build varies kinds of applications in the industry. Some examples are sentiment analysis, chatbot and virtual assistants, text classification, text extraction, machine translation, text summarization, market intelligence, auto-correct, intent classification, urgency detection, and speech recognition. Related with content-based recommendation, the suitable NLP task is a document similarity calculation, where each document has a similarity score compared to other document. If a user accesses a certain document, the system will recommend other documents with highest content-based similarity score with the current document being read by user.

The common technique to calculate the similarity between two documents is vector space model (Salton, et.al, 1975). Here, each document is represented as a vector of numbers, where the number represent the score of a word weight. The equation of a document vector (Vd) is shown below.

$$V_d = [w_{1,d}, w_{2,d}, \dots, w_{n,d}]^T$$

Each document vector (Vd) consists of an n number of word weight (from w1,d to wn,d). Here n is the dimension of the document vector and it represents the number of unique words in the document collection. Number of unique words in document collection or called as vocabulary size is a lot more higher that number of unique words in certain document. Thus, it causes the document vector as a sparse vector containing many zero scores.

There are several alternatives in defining the score in the word weight (from w1,d to wn,d). It can be a boolean score showing a word existence in a document, zero if the wordi doesn’t exist in document d or one if the wordi exists in document d. It can also be a term frequency (TF) score, where the score wi,d represents the number of wordi found in document d. It can also be a TFxIDF score (Salton, et.al, 1975) where the score wi,d represents the number of wordi found in document d divided by the number of documents (df) containing wordi.

The similarity score between two documents is the distance score between two document vectors. If there is no similar words between two given documents, then the similarity score will be zero. The common distance score is cosine similarity score, which is a dot product between two vectors divided by the product of vector length. The equation of cosine similarity score is shown below.

$$\cos(d_j,d_k) = \frac{d_j.d_k}{\|d_j\|\|d_k\|} = \frac{\sum_{i=1}^{n} w_{i,j}w_{i,k}}{\sqrt{\sum_{i=1}^{n} w_{i,j}^2}\sqrt{\sum_{i=1}^{n} w_{i,k}^2}}$$

The dot product of two document vectors is the cumulative of the product between two word weights (wi,j and wi,k) which represents the wordi weight in each of two documents (dj and dk). The result of the dot product of two document vectors is then divided by the product of length of dj vector and length of dk vector.

IV. Reinforcement Learning

Reinforcement Learning is a machine learning algorithm using reward signal to indicates whether an action of an agent is good or bad. It doesn’t receive perfect feedback such as in supervised learning. Taking the markov model of an intelligent agent, a reward of a state (given an action) will affect the score of the state. The description of a reinforcement learning of an intelligent agent is shown in figure below.

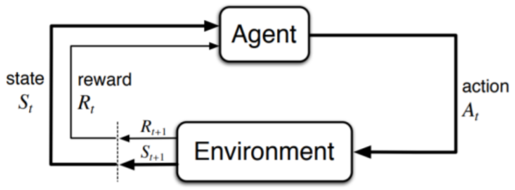

Figure 4.1. Illustration of Intelligent Agent with Reward (Sutton&Barto, 2018)

An agent will select an action At based on the agent state (St) and the reward (Rt) received in the state St. The action At will cause a new state (St+1) with its reward signal (Rt+1). The previous state, reward and action will influence the next action selected by agent and the state + rewards selected by the environment. The agent goal is to take actions that maximize the cumulative rewards. Basically there are several types of reinforcement learning agent, related with its behaviour (known as policy), value function (prediction of future reward) and model (a probability score given by environment about the next state and next reward signal). The most common type is to use only value function in selecting best actions. A simple algorithm using only value function to make optimal policy is TD Q-learning. In TD Q-learning, the action value function (known also as q-function) of a given state and given action is updated by the reward and difference between current state and next state. The complete equation is shown below.

$$Q(S,A) = Q(S,A) + \alpha[R + \gamma \max_a Q(S',a) - Q(S,A)]$$

Each state and action has its action value function (Q(S,A)). This value function is updated using learning rate (α) multiply with the accummulation of Reward(R) added by the difference between the maximum action value function of next state (maxa Q(S’,a)) and the current state given action(Q(S,A)). The difference is multiply first by a discount factor (γ). A sequence of state and action from start until finish is called one episode. In order to achieve best result, there should be many episodes conducted. In each episode, the action value function will be updated and the agent will give optimal policy based on the updated action value function.

V. Proposed Method

One method which proposed to a recommendation system is by using weight-directed graph and reinforcement learning. This simple model is built with several orders of action which will be explained below.

1. Graph Initialization

The first step is to build a correlated graph represent the knowledge data, which are the songs alongside with the attributes. As mentioned in the previous section, a graph is assembled by nodes and edges. In this model, a node represents a song along with its properties or attributes. Attributes descripted from each songs are:

1. Title,
2. Singer,
3. Genre,
4. Lyrics,
5. Accessed boolean score.

Each of the attribute is acquired from input songs data. These data then will be processed into machine learning algorithm called Natural Language Processing or shortened as NLP, to count the similarity number of each node as the base for building the weight-directed graph.

A node has the representation of a song, while an edge has the representation of the relation between songs. 'Related' songs are connected with the same edges, thus given the direction to specify which song is relevant and similar to another.

Given the illustration of the graph below. Let's say we have the data of 3 songs with attributes and relation, but have not had any weight on each edge. The graph is initialized as a directed graph, with node 1 correspond to song 1, node 2 correspond to song 2, and node 3 correspond to song 3.

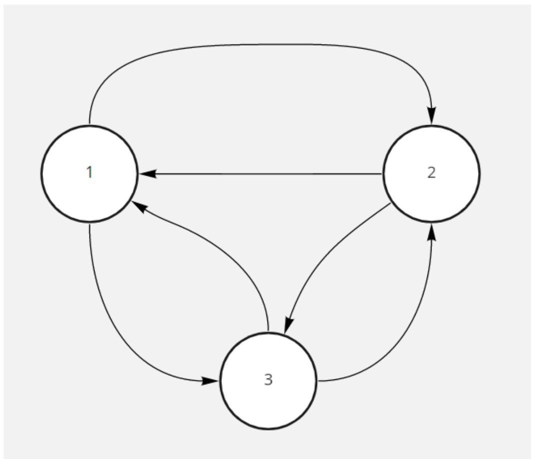

Fig 5.1 Graph Initialization

Each song has its own data of title, singer, genre, and lyrics which will be the inputs for the algorithms. After building the initialization graph, we are going to add the weights for each edges. The initialization weight of the edges are counted from the similarity of the singer, genre, and lyrics which are calculated by the algorithms of NLP.

1.1 Weight Initialization with NLP Algorithm

The common technique to calculate similarity between two documents is vector machine model. Each document of a song is represented as vector of number, which is equated in a form of vector below.

$$V_d = [w_{1,d}, w_{2,d}, \ldots, w_{n,d}]^T$$

Then, each pair of song will have the number of cosine similarity, based on the proximity of the documents including titles, lyrics, and singers. The equation of cosine similarity score is shown below.

$$\cos(d_j,d_k) = \frac{d_j . d_k}{\|d_j\|\|d_k\|} = \frac{\sum_{i=1}^{n} w_{i,j} w_{i,k}}{\sqrt{\sum_{i=1}^{n} w_{i,j}^2}\sqrt{\sum_{i=1}^{n} w_{i,k}^2}}$$

Each of the attributes have their own similarities. Thus, these numbers are generated into the weight formula with different percentages. In this models, we will use the formula to count each of edge weights.

Edge weight = 0.5*(singer similarity) + 0.3*(genre similarity) + 0.2*(document similarity)

The word 'document' in the formula is represented as the song title along with the lyrics correspond. While the percentage above is determined by several consideration. Users are tend to like the songs from the same singer, so that it has the highest number of percentage. Songs with the same genre also tend to be loved by the same user, so it comes with second rank of percentage which is 30% or 0.3. Lastly, the topic of the songs which known from the similarity of the lyrics and title are chosen as the lowest percentage.

After the edge weighting, the graph will be updated as below.

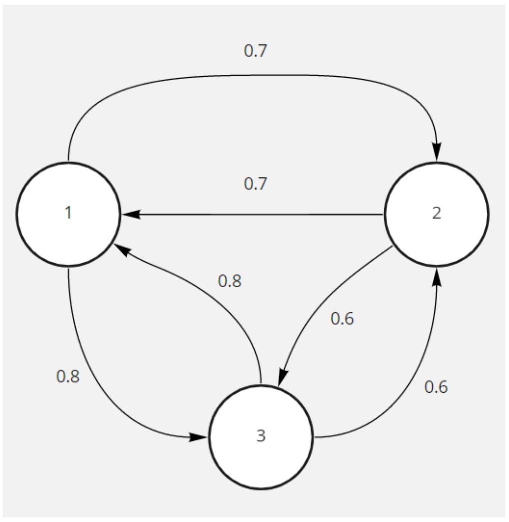

Fig 5.2 Weight-Directed Graph

Each pair of node is connected by 2 edges with different direction, but weights the same amount, because each node has the same similarity value. For example, song 1 has 0.7 similarity towards song 2, and vice versa.

2. Update Edges' Weights

User will be recommended the next song as the arrow direction follows. There will be two types of actions from users when they got the next song as heard in the queue playlist. Users will let the song played if they like it, and they will skip the song if its unwanted. So that we defined the actions from users are skip and listen.

These input data will then be calculated in the reinforcement learning algorithm using equation below.

$$Q(S,A) = Q(S,A) + \alpha[R + \gamma \max_a Q(S',a) - Q(S,A)]$$

From the equation, we defined **S** as the song which has just being heard or recommended by the system, **a** as the action inputs by user which includes skip and listen, **α** which is chosen as the small default number by 0.2 because of the small differences between recommended songs, **R** which is the reward value determined from the user's action ('skip' action will be valued as -10, and 'listen' action will be valued as 10), **γ** as the discount factor which is default by 0.5 because the importance of the next state is half of the importance of the current state.

The value of accessed boolean value in each song nodes will be updated on each episode, where episode is started from the first song until he/she finishes listening to songs following by closing the app. The accessed boolean value of the song will be marked as 0 if the song hasn't heard or recommended to the user, and will be marked 1 if the song has been given or recommended to the user.

Let's say the user has given the first song, which is song 1. After listening through the first song, the system gives the song 2 as the next song recommendation. The user seems to unlike the chosen song, so that he/she gives the 'skip' action, resulting in his/her getting the second song recommendation which is song 3. The weight of the edge from song 1 to song 2 will be decreased since the reward is negative. Thus, the accessed boolean value for song 1 and song 2 are updated as 1. Q value function for each state and action is updated every time the user take an action towards the recommended song. The figure below has shown the updated weight-directed graph as the user take a 'listen' action towards the song 3.

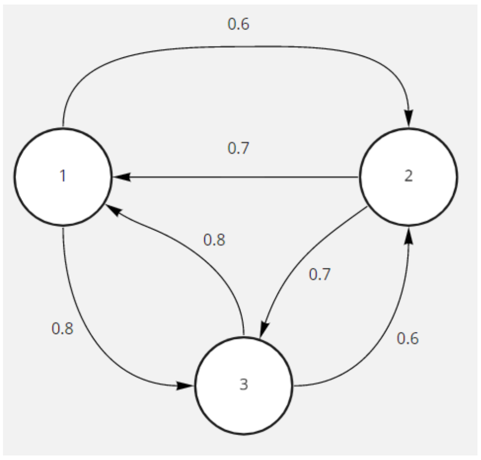

Fig 5.3 Updated Graph

The value of edge from song 1 and song 2 is decreased, while the value of edge from song 2 and song 3 is increased.

VI. CONCLUSION

Weigh-directed graph can be used as the data structure for the recommendation system with varies types of data. This paper examine the recommendation system method for songs, with the addition application of reinforcement learning algorithm. The nodes for graph based song recommendation system include information about the title, singer, genre, lyrics, and accessed boolean score, while the edge represent the recommendation score of each song which is initialized by the similarities of singer, genre, titles, and lyrics. As the user got the song recommendation, he/she could take 2 types of action (skip and listen). The action will effect the recommendation score as it updated the weight of the edge correspond. Those updating system on the recommendation score follows the reinforcement learning algorithms where positive reward is given to 'listen' action, and negative reward is given to 'skip' action. For the next episodes of listening, user will have new and updated song recommendation graph.

VII. ACKNOWLEDGMENT

The author would like to thank first of all, to God for all the guidance thoughout the process of learning thus writing the contents of this paper. The author would also like to thank the lecturer of ITB Discrete Mathematics IF2120, Mrs. Ulfa Nur Maulidevi, Mrs. Harlili, Mrs. Fariska Zakhralativa, and Mr. Rinaldi Munir for sharing their knowledges and guide the students throughout the learning process in the class. Not to forget, the author would also like to thank her family and friends who have given the support throughout the entire semester..

REFERENCES

[1] G. Salton , A. Wong , C. S. Yang, A vector space model for automatic indexing, Communications of the ACM, v.18 n.11, p.613–620, Nov. 1975W.-K. Chen, *Linear Networks and Systems* (Book style). Belmont, CA: Wadsworth, 1993, pp. 123–135.
[2] Richard S. Sutton & Andrew G. Barto, Reinforcement Learning “An Introduction”, MIT Press, 2018B. Smith, “An approach to graphs of linear forms (Unpublished work style),” unpublished.
[3] Aggarwal, Charu C. (2016). Recommender Systems: The Textbook. Springer. ISBN 9783319296579..
[4] https://newsroom.spotify.com/2020-11-02/amplifying-artist-input-in-your-personalized-recommendations/
[5] https://mixpanel.com/topics/what-is-user-engagement/
[6] https://www.researchgate.net/publication/298801724_Survey_on_Recommendation_System
[7] https://medium.com/s/story/spotifys-discover-weekly-how-machine-learning-finds-your-new-music-19a41ab76efeJ.
[8] Munir, Rinaldi.”http://informatika.stei.itb.ac.id/~rinaldi.munir/”

PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 11 Desember 2020

[illegible]

Aulia Adila - 13519100