# [AI506] Data Mining - Progress Report

Sungjoo Han
Bio and brain engineering, KAIST
Daejeon, South Korea
sungjoohan@kaist.ac.kr

Dayeon Jeong
Bio and brain engineering, KAIST
Daejeon, South Korea
doremikong@kaist.ac.kr

## ABSTRACT

The fashion industry is constantly changing, with new trends and styles emerging every season. However, many people struggle to put together outfits that are both fashionable and comfortable. To address this issue, this paper presents a system for personalized outfit recommendations and outfit generation using data mining techniques. The system employs three methods: collaborative filtering, node embedding, and association rule mining. Collaborative filtering helps to identify users' preferences for specific items based on the preferences of similar users and items. Node embedding uses the Node2vec algorithm to learn embeddings of outfits and user preferences in a high-dimensional space, enabling the system to recommend outfits that are similar to those that other users with similar preferences have interacted with. Association rule discovers relationships and correlations between items in datasets and generates a missing item for incomplete outfits. The proposed system aims to create outfits that are both aesthetically pleasing and tailored to each user's individual style preferences with high accuracy and efficiency. Overall, the system offers a useful tool for fashion-conscious individuals who want to effortlessly create stylish outfits that reflect their personal style preferences.

## 1 INTRODUCTION

The fashion industry is constantly evolving, with new trends and styles emerging every season. However, for many people, putting together a stylish outfit that looks great and feels comfortable can be a challenge. This is where outfit generation and recommendation systems come into play, which use data mining techniques allowing for the analysis of large amounts of data to identify patterns and relationships between items.

There are two main tasks in this study. First task is to present personalized outfit recommendations for users and the second task is to generate a new item for incomplete outfits. By analyzing users' interaction information with individual items and outfits, the system can learn their style preferences and the concept of each outfits so that it generates outfits that are tailored to each users' individual tastes and each outfit.

To address the problems, we propose three methods: collaborative filtering, node embedding, and association rule.

- Collaborative filtering: It finds user's preference for a particular item based on the preferences of similar users and items. Using collaborative filtering, we found similar users and itemsets to classify personalized outfit.
- Node embedding: Node2vec algorithm is used to learn embeddings of outfits and user preferences in a high-dimensional space. These embeddings are then analyzed to find similar users and similar item sets so that we can recommend the users the outfits which have an interaction with similar users.
- Association rule: It discovers relationships and correlations between items in datasets. It works by identifying frequently co-occurring items and generating rules that describe the relationships between them. Based on the generated association rule, candidate items for incomplete outfits could be found.

Overall, our goal is to provide a valuable tool for fashion-conscious individuals who want to effortlessly put together stylish outfits that reflect their personal style preferences. By leveraging the power of node2vec and association rule, we hope to create a system that not only generates aesthetically pleasing outfits but also meets users' needs and preferences with high accuracy and efficiency.

## 2 EXPLORATORY DATA ANALYSIS

### 2.1 Task 1: Outfit recommendation

Task 1 training set contains the users and corresponding itemsets. We analyzed the data in the aspects of users and itemsets respectively. Table 1 shows the reult of statistical analysis in the training set. In addition, Figure 1, 2 show the distribution of each of them.

| | Itemsets per users | Users per itemsets |
|---|---|---|
| Mean | 24.9 | 48.5 |
| Standard divation | 14.6 | 97.2 |
| Min | 7.0 | 2.0 |
| 25% | 18.0 | 11.0 |
| 50%(Median) | 21.0 | 20.0 |
| 75% | 27.0 | 44.0 |
| Max | 472.0 | 2995 |

**Table 1: Statistics of task 1 training**

To examine the user perspective, the number of item sets utilized by each user was analyzed. On average, users were found to employ approximately 24.9 item sets. The user with the fewest item sets had only 7, while the user with the highest number of itemsets had 472. Although the maximum value is significantly high, the majority of users (from the 1st quartile to the 3rd quartile) had between 18 and 27 item sets (Figure 1). For further analysis, a threshold of approximately 25 item sets per person (based on the mean value) will be set as hyperaparmeter in the consturction of model.

From the perspective of item sets, an analysis was conducted to determine the number of users associated with each itemset. On average, each item set was found to be possessed by approximately 48.5 users, with a significantly large standard deviation of 97.2 users. The itemset held by the fewest number of users was associated with only 2 individuals, while the itemset with the highest number of

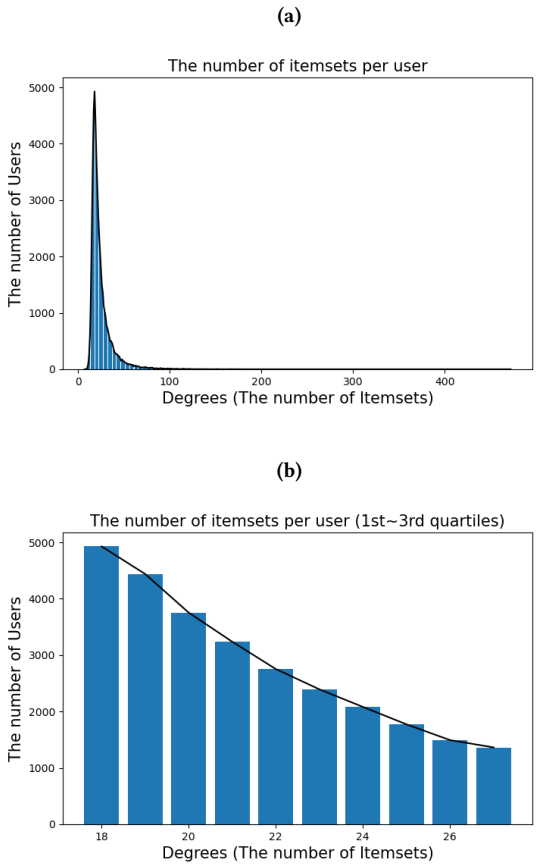


**Figure 1: (a) The number of itemsets per users (b) The number of itemsets per users (1st-3rd quartiles)**

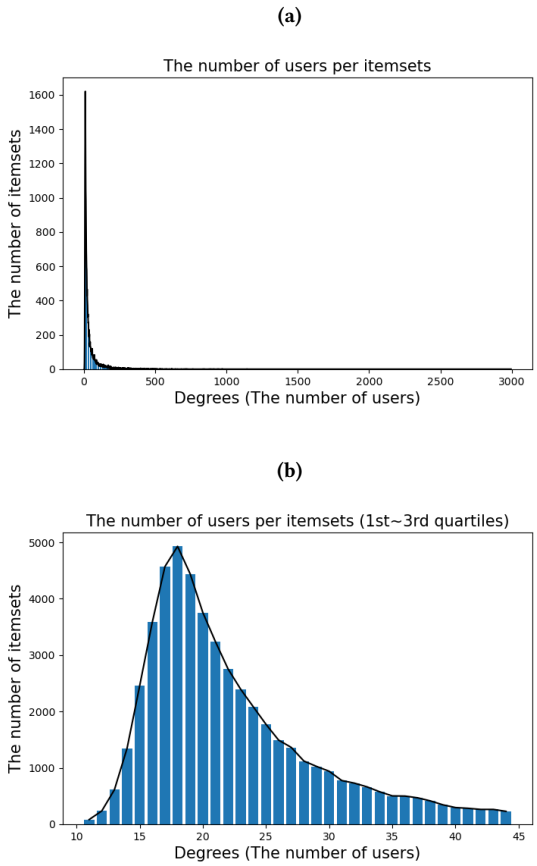


**Figure 2: (a) The number of users per itemsets (b) The number of users per itemsets (1st-3rd quartiles)**

users was associated with 2,995 individuals. However, the majority of item sets were utilized by a range of 11 to 44 users (Figure 2). For further analysis, a threshold of approximately 50 users per itemsets (based on the mean value) will be set as hyperaparmeter in the consturction of model.

## 2.2 Task 2: Missing Item Generation

Task 2 training set contains the itemsets and corresponding items. We analyzed the data in the aspect of itemsets and items respectively. Table 2 shows the reult of statistical analysis in the training set. In addition, Figure 3 shows the distribution of each of them.

First, from the perspective of item sets, an analysis was conducted to check the number of items per itemsets. As shown in Figure 3 and Table 2, we could check all itemsets contain 3, 4, or 5 items. From the perspective of items, an analysis was conducted to determine which item data to use for association rule mining. We analyzed the incidence count of each items in training dataset. As shown in Figure 3 and Table 2, most of items are presented only once. Therefore, items with incidence count 1 were removed when generating association rule so that we can efficiently deal with large data.

| | Items per itemsets | Itemsets per items |
|---|---|---|
| Mean | 2.02 | 3.87 |
| Standard divation | 4.5 | 0.77 |
| Min | 1 | 3 |
| 25% | 1 | 3 |
| 50%(Median) | 1 | 4 |
| 75% | 2 | 4 |
| Max | 285 | 5 |

**Table 2: Statistics of task 2 training**

# 3 PROPOSED METHOD

## 3.1 Problem Setup

The goal of the outfit recommendation is to judge whether the user $\mathbf{x} \in \mathcal{X}$ prefers the given outfit (i.e., itemsets) $\mathbf{y} \in \mathcal{Y}$ where $\mathcal{X}$ is the set of users and $\mathcal{Y}$ is the set of outfits.

The goal of the outfit generation is to predict the missing fashion item $y \in \mathcal{Y}$ for the given incomplete outfits (i.e., itemsets) $\mathbf{x} \in \mathcal{X}$ where $\mathcal{Y}$ is the set of items and $\mathcal{X}$ is the set of outfits.

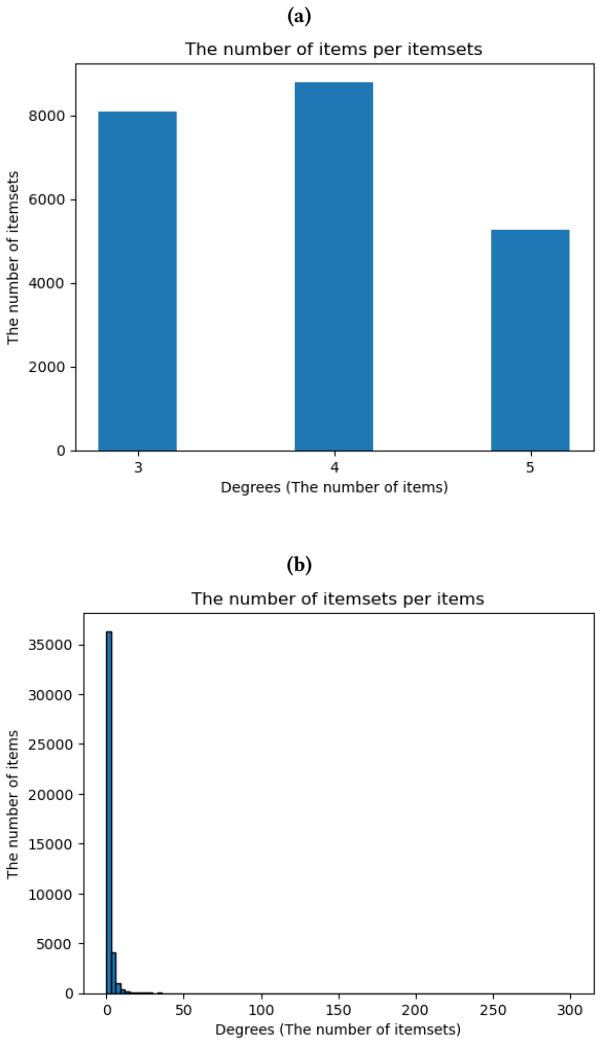


**Figure 3: (a) The number of items per itemsets (b) The number of itemsets per items**

## 3.2 Base algorithms

We used 3 base algorithms for two tasks.

*i) Collaborative filtering.* It is a technique used in recommendation systems to provide personalized recommendations to users based on their preferences and behaviors. The basic idea behind collaborative filtering is to analyze the patterns and relationships among users and items to make predictions about a user's interests or preferences. Collaborative filtering algorithms use this user-item matrix to identify similar users or items based on their rating patterns. There are two main types of collaborative filtering techniques.

User-based collaborative filtering is calculated as: (1) similarity between users i and j using cosine similarity and (2) prediction of user i's rating for item k using weighted average.

$$\text{sim}(u_i, u_j) = \frac{\sum_k r_{ik} \cdot r_{jk}}{\sqrt{\sum_k r_{ik}^2} \cdot \sqrt{\sum_k r_{jk}^2}} \tag{1}$$

$$\hat{r}_{ik} = \frac{\sum_{u_j} \text{sim}(u_i, u_j) \cdot r_{jk}}{\sum_{u_j} |\text{sim}(u_i, u_j)|} \tag{2}$$

where $\text{sim}(u_i, u_j)$ is the similarity between users i and j, $r_ik$ is the rating of user i for item k, $r_jk$ is the rating of user j for item k, $\hat{r}_{ik}$ represetns the predicted rating of user i for item k.

Item-based collaborative filtering is calculated as: (3) similarity between items i and j using cosine similarity and (4) prediction of item i's rating for user k using weighted average.

$$\text{sim}(i, j) = \frac{\sum_u r_{ui} \cdot r_{uj}}{\sqrt{\sum_u r_{ui}^2} \cdot \sqrt{\sum_u r_{uj}^2}} \tag{3}$$

$$\hat{r}_{ik} = \frac{\sum_{u_j} \text{sim}(i, j) \cdot r_{uj}}{\sum_{u_j} |\text{sim}(i, j)|} \tag{4}$$

where $\text{sim}(i, j)$ is the similarity between item i and j, $r_ui$ is the rating of user u for item i, $r_uj$ is the rating of user u for item j, and $\hat{r}_{ik}$ represetns the predicted rating of item i for user k.

*ii) Node2Vec.* It is an algorithmic framework for representational learning on graphs. The goal of Node2Vec is to capture the structural properties of a network and encode them into low-dimensional vectors, which can be used for various downstream tasks such as node classification, link prediction, and recommendation systems.

The Node2Vec algorithm involves a two-step process: (5) generating biased random walks on the network, and (6) using these random walks to train a Skip-gram model similar to Word2Vec. The biased random walks aim to explore both the local and global neighborhood of each node, capturing both the community structure and the structural equivalence between nodes.

$$P(v_j \mid v_i) = \begin{cases} \frac{1}{p} & \text{if } (v_i, v_j) \in E \\ 1 & \text{if } (v_i, v_j) \notin E \\ \frac{1}{q} & \text{if } (v_i, v_j) \notin E \text{ and } v_j \neq v_i \end{cases} \tag{5}$$

$$\max \sum_{v_i \in V} \sum_{v_j \in \mathcal{N}(v_i)} \log P(v_j \mid v_i) \tag{6}$$

where $v_i$ is a given node, $v_j$ is a neighbor node, $\notin E$ represents the set of edges in the network, $p$ and $q$ are turning parameters that control the likelihood of exploring local versus global neighborhoods, and $V$ represents the set of all nodes in the network.

*iii) Association rule.* It is a technique used to discover interesting relationships or associations between items in a large dataset. The most commonly used measure for association rule mining is called support, and confidence. Support measures the frequency or occurrence of an itemset in the dataset. It indicates how often an itemset appears in the dataset. Confidence measures the likelihood of item B appearing in a transaction given that item A is present in that transaction. It indicates the strength of the association between items A and B.

The association rule is typically written in the form of $(A \rightarrow B)$, where A is the antecedent (itemset) and B is the consequent (itemset). the number of transactions containing A refers to the count of transactions in which the itemset A is present, and the total number of transactions represents the overall count of transactions in the dataset. The $\sup(A \cup B)$ indicates the number of transactions in which both A and B are present.

$$\sup(A) = \frac{\text{number of transactions containing A}}{\text{total number of transactions}} \tag{7}$$

$$\text{conf}(A \rightarrow B) = \frac{\sup(A \cup B)}{\sup(A)} \tag{8}$$

where itemset A and itemset B.

## 3.3 Task1 Model Overview

Figure 4 shows the overview of the task 1 model. To address Task 1, we independently implemented two approaches: collaborative filtering and ndode embedding. The selection of the final model will be based on its superior performance. Both methods aim to identify similar users or item sets and make predictions. However, collaborative filtering utilized Pearson correlation on a one-hot matrix, while node embedding employed the Node2vec algorithm, which simulated random walks on a graph and captured both local and global structural information.

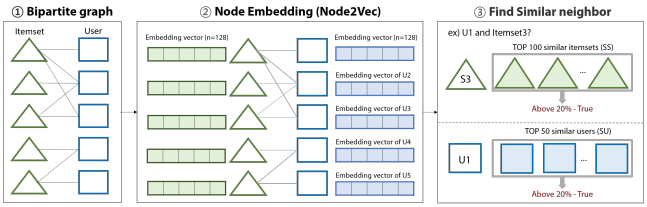


**Figure 4: Task1 overview**

*i) Collaborative filtering.*

- Construction of one-hot matrix: A one-hot matrix was created, with users represented as rows and item sets as columns. The train dataset was utilized to establish the user-item set relationships, encoding them as "1" in the matrix.
- Calculation of Pearson Correlations: Pearson correlation coefficients were computed to determine the similarity between users (user-user) and item sets (item set-item set). This involved evaluating the pairwise correlations among the rows (users) and columns (item sets) of the one-hot matrix.
- User-user prediction and itemset-itemset prediction: For user-user prediction, it was conducted by selecting the top 100 users with the highest similarity scores. The prediction involved multiplying the similarity score by the item set value (0 or 1) of y. For itemset-itemset prediction, the top 100 item sets with the highest similarity scores were chosen. The prediction entailed multiplying the similarity score by the user value (0 or 1) of x.

*ii) Node embedding using Node2vec.*

- Construction of bipartite graph: A bipartite graph was constructed using the user-item set dataset. This graph consisted of two types of nodes: users and item sets.
- Node2vec algorithm for node embedding: The node2vec algorithm was employed to generate node embeddings for each user and item set in the bipartite graph. We embedded node into 128-dimensional vector and set 10 for walk length and 30 for number of walks as hyperparameters.
- Identification of similar nodes: To identify nodes similar to the target (user, itemset), a similarity analysis was conducted. The twice average value (obtained from statistical analysis) as a threshold, and the top 50 nodes similar to the target user and the top 100 nodes similar to the target item set were identified.
- Classification based on node inclusion: Among the total of 150 nodes identified in step 3, if 10%, 20%, or 30% of these nodes included the target user or target item set, they were classified as "1," indicating inclusion.

## 3.4 Task2 Model Overview

Figure 5 shows the overview of the task 2 model. For the missing item generation task, the association rule is used. Especially, Apriori algorithm is used because of its efficiency and scalabilty which makes it suitable for large datasets. Once the association rule is generated, 100 items for each itemset are recommended based on the association rule.

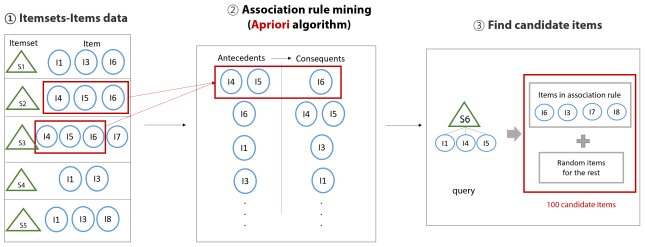


**Figure 5: Task2 overview**

*i) Association rule mining using Apriori.*

- Construction of one-hot matrix: The itemset-item dataset was converted to a One-Hot Encoded Boolean list. Items that the outfits contain or not are represented by values True and False. Each row corresponds to the outfit and each column corresponds to the item.
- Apriori algorithm for association rule mining: The Apriori algorithm was employed to discover frequent itemsets and generate association rules. These rules capture the relationships between items in the dataset. By setting the minimum support, it uses only frequent items.
- Ranking candidate missing items: We rank 100 candidate missing items for each itemset query based on the generated association rule. For each item in query, we check if the item is in the antecendents of the generated association rule. If it is, then we make the consequents of that antecendents

| Method | Hyperparameter | Accuracy |
|---|---|---|
| Random guessing | - | 0.500 |
| Frequent itemsets | - | 0.615 |
| Collaborating filtering | user-user prediction | 0.479 |
| | itemset-itemset prediction | 0.475 |
| Node2Vec | 10% | 0.676 |
| | 20% | **0.702** |
| | 30% | 0.636 |
| NodeVec-Neighbor | - | 0.636 |

**Table 3: Result of Task1**

| Method | Accuracy | Rank |
|---|---|---|
| Random guessing | 0.001 | 100.955 |
| Association rule mining (Apriori algorithm) | **0.101** | **93.787** |

**Table 4: Result of Task2**

be candidate items. After checking all the association rules, we fill the rest of the candidate items by random until the number of the candidates become 100.

# 4 EXPERIMENTS

## 4.1 Task1 experiment

Table3 shows the result of task1. The models based on the Node2vec algorithm showed the highest accuracy, ranking 1st to 3rd. Specifically, the model utilizing a 20% hyperparameter exhibited the best performance. In contrast, the models utilizing collaborative filtering showed lower performance compared to the models employing random and frequent methods. The performance of the models utilizing user-user interaction and item set-item set interaction was similar.

## 4.2 Task2 experiment

Table4 shows the result of task2. The model based on the association rule mining with Apriori algorithm showed the accuracy of 10.1% and mean rank 93.787. The accuracy was almost hundred times greater than the accuracy of random guessing.

# 5 CONCLUSIONS

For personalized outfit recommendations, we explored two methods: collabroative filtering and node embedding. We found the effectiveness of the Node2vec-based model, particularly when utilizing a 20% hyperparameter, while suggesting limitations in the collaborative filtering approach and comparable performance between user-user and item set-item set interactions.

For missing item generation, we used association rule mining with Apriori algorithm. By finding the frequent itemsets in data with association rule, the candidate items were recommended for each itemset. We found that the association rule mining approach showed much better performance than random guessing, and therefore found the effectiveness of this approach.

Overall, we suggest that Node2vec-based model and Apriori-based model are effective and could be useful tools for recommendation system especially in fashion industry. Further, these models might also be useful for any recommendation systems not only for fashion recommendation system because of their ability to find the similarity and rules in large datasets.

# A APPENDIX

## A.1 Labor Division

The team performed the following tasks

- Overall discussion of the method [all]
- Implementation of Task1 model [Han]
- Implementation of Task2 model [Jeong]

## A.2 Full disclosure wrt dissertations/projects

*Han:* She is not doing any project or dissertation related to this project: her thesis is on drug recommendation using EHR analysis.

*Jeong:* She is not doing any project or dissertation related to this project: her thesis is on drug-target discovery.