Chapter 0

Invitation: Recommender System

A recommender system that seeks to predict the "rating" or "preference" that a user would give to an item. It has become increasingly popular in recent years, and are utilized in a variety of areas including movies, music, news, books, research articles, search queries, social tags, and products in general. There are also recommender systems for restaurants, garments, financial services, life insurance, romantic partners (online dating), and Twitter pages.

Suppose we found a startup that provides online movie viewing. Our company has attracted $i = 1, 2, ..., m$ subscribers, and has acquired $j = 1, 2, ..., n$ movies. Each subscriber is invited to rate a movie from $r = 1, 2, .., 5$, after viewing the movie, These ratings are stored in a $m \times n$ sparse matrix $R(i, j) = r_{i,j}$, containing records of each subscriber's ratings on all movies in our database.

Note that the rating remain at $r = 0$ for movies that the subscriber hasn't watched or rated. Based on this existing data, we want to build a recommender system to learn about the "taste of the subscriber, so that we can make predictions on a particular subscriber's ratings of these movies that he/she has not watched.

In a method known as collaborative filtering, each movie is evaluated against certain general features, labelled as $k = 1, 2, .., p$, that describe movies. Typically, only a few features are needed and $p \leq n$ (provided our company acquires a lot of movies). These "hidden" features, which arise in our implementation algorithm, depend entirely on the data, and are not necessarily related to typical classifications such as comedy, tragedy, actions etc. More concretely, for a

particular movie j', we are to evaluate a series of scores $y_{j',k}$ against all hidden features $k = 1, 2, .., p$ that best describe the movie. The collection of all such scores is to be stored in the $n \times p$ matrix $Y(j,k)$, which we are to calculate.

On the other side, these hidden features can also be used to describe the subscribers' preferences on movies. For a subscriber i', we want to evaluate a series of scores $x_{i',k}$ against all hidden features $k = 1, 2, .., p$ that best describe the subscriber's taste on movies. These scores are to be collectively stored in the $m \times p$ matrix $X(i,k)$, which are also to be found.

Now our task is to find the matrices $X(i,k)$ and $Y(j,k)$, which amount to $(m+n) \times p$ unknown parameters. The product matrix, $X \cdot Y^T = \sum_{k=1:p} X(i,k) \cdot Y(j,k)^T$ of $m \times n$ dimensions can be compared to the subscriber's rating $R(i, j)$. While $R(i, j)$ is sparse (since each subscriber only had viewed or rated a small amount of the entire movie database), our algorithm is to minimize the difference between the known (non-zero) ratings in $R(i, j)$ and the corresponding rating predicted on $\sum_{k=1:p} X(i,k) \cdot Y(j,k)^T$. In other words, we are to minimize the error function, defined as:

$$J(x_{i,k}, y_{j,k}) = \sum_{\substack{i=1:m,\, j=1:n \\ R(i,j) \neq 0}} [\sum_{k=1:p} X(i,k) \cdot Y(j,k)^T - R(i, j)]^2 \ ,$$

for $i = 1, 2, .., m$, $j = 1, 2, .., n$, $k = 1, 2, .., p$, and $p \leq m, n$

Optimization scheme are used to find the set of parameters $x_{i,k}$ and $y_{j,k}$ for all $i = 1, 2, .., m$, $j = 1, 2, .., n$ and $k = 1, 2, ..., p$ that gives the global minimum of this error function.

Upon this optimization procedures that minimize J, both the subscribers' tastes on movies $X(i,k)$, and the nature of movies $Y(j,k)$ can be estimated. The $m \times n$ matrix $X \cdot Y^T \big|_{\min(J)}$ then stores the predicted ratings of all movies to all subscribers. For instance, the i'-th row of $X \cdot Y^T \big|_{\min(J)}$ is a row vector that lists the predicted ratings of all movies $j = 1, 2, ..., n$ to that i'-th subscriber. These predicted ratings of $X \cdot Y^T \big|_{\min(J)}$ should be nonzero on those (i, j) entries with unknown ratings, where $r_{i,j} = 0$, while on those (i, j) entries where $r_{i,j} \neq 0$, should roughly be equaled to the known ratings of $R(i, j)$. By sorting these ratings in descending order, our system should be able to show a subscriber our top recommendations that best match to his/her taste.