

MATH 482

Matrix Factorisation Project

Daniel Braithwaite

May 18, 2017

1 Introduction

To discuss the idea of matrix factorisation and methods to solve it first we must understand the motivation for wanting to solve such a problem. In the case of the Netflix challenge the problem was to build a system to recommend movies to users. We have this very large matrix R with the rows corresponding to a user and a column corresponding to a movie. The entry $R_{i,j}$ is the rating that user i gave movie j , in practice we would find that a very small percentage of this matrix would be filled in. To make recommendations we would like to predict the ratings which a user might give a movie which they haven't watched.

2 Solution 1: $R = U * M$

The first solution we consider is that R (an uxm matrix) is actually the product of two smaller matrices U and M . Where U (a uxk matrix) represents the users in some latent feature space and M (a mxk matrix) represents the movies in the latent feature space. We consider $M_{i,j}$ to be the amount movie i has feature j , likewise we consider $U_{i,j}$ to be how much user i is interested in movies with feature j . Then we can take the rating user i gives movie j to be $R_{i,j} = \text{row}(U, i)^T \cdot \text{row}(M, j)$. Now the problem becomes how do we learn these matrices U and M .

We consider the following optimization problem, where G contains all pairs (i, j) for which we know $R_{i,j}$

$$\min_{U, M} \sum_{(i,j) \in G} (R_{i,j} - \text{row}(U, i)^T \cdot \text{row}(M, j))^2$$

This optimization problem can be solved with gradient descent