Algorithms

Factorization

See machine learning video

SVT

FPCA

Optspace

## Matrix completion by factorization

Matrix completion by factorization is not a very good algorithm in the sense that it is a heuristic and proper mathematical analysis is not available very clearly in literature. However this has been applied by Simon Funk successfully in Netflix prize, it also works for a variety of datasets and do so quite fast. This is a variant of simon-funk [19] type heuristic with some modifications in the algorithm.

The user-movie dataset matrix M can be thought to be made up of two latent feature matrices U and V which are sampled from normal distribution for initial value. These matrices are updated in the loop and with each iteration it tries to get closer to original matrix.

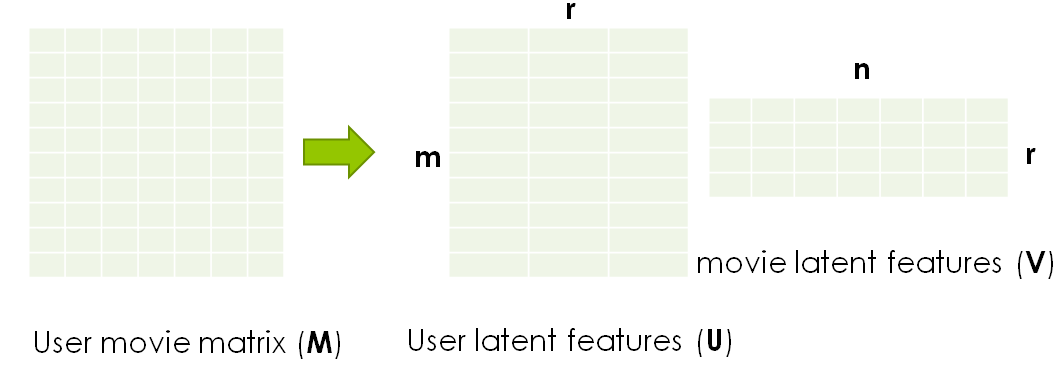


Figure 4.2 – Data matrix as two latent feature matrices

The original matrix M is taken to be a binary matrix containing 0 and 64 as depicted in the figure for simplicity and studying the matrix completion process.

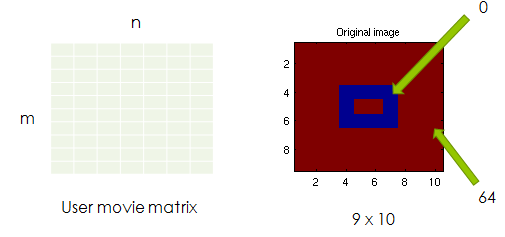


Figure 4.3 - Original matrix M

To study the matrix completion we now randomly hide some entries from the matrix and these missing locations are itself chosen from a normal distribution of internes ranging from 1 to n\*m without replacement so that values does not repeat. Hence by following such a procedure we can assure ourselves that the given number of entries is missing without any doubt.

**Code**



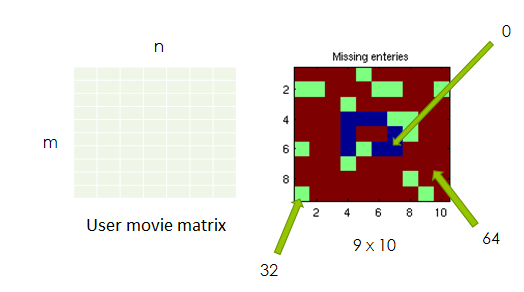


Figure 4.4 - Matrix with missing enteries M\_missing

The values shown in blue are 0 and that shown in red are 64 as in the figure above, however the entries in green are missing and their information will not be used in the algorithm which predicting the original matrix. We have used a value of 32 just for illustration purposes and to depict it clearly, The value assigned to missing numbers does not matter as it is not to be used in the algorithm which computing the original matrix. The assumption that it is of low rank holds good here. The goal of our algorithm is that we wish to recover the original matrix from this matrix with missing enters.

### Mathematical model

The goal of the algorithm is to minimize the frobinius norm between the two matrices X and M.

Or equivalently

We update matrix x so that our error term e gets minimised after each iteration, a very simple algorithm for this is introduced by Simon Funk here [19]. The algorithm here is gradient descent, so calculate the delta term by differentiating the error term as under

Decompose the matrix into two latent factors as described above





This is the error with respect to each entry between the two matrices.



We differentiate this error with respect to corresponding row and column (latent features here) to get update rules for gradient descent and add a learning rate α.





### Algorithm

With the values of U and V we calculate the values of M’ after each iteration and compare it with M. The algorithm updates the rows and columns of U and V so that the norm of the residual is minimised. We found that such a direction is given by steepest descent algorithm.

* Initialize the matrices U and V or S and R.
* For k = 1 to max\_iter
  + Loop
    - Loop for all
      * Compute error
      * Update features
  + Recompute and print error term
* end

Here is a snippet of MATLAB implementation of the algorithm above.

