Personal data

Factorization of the user-item matrix

Let us consider a matrix M indicating numerical evaluations provided by 100 people on 1000 items. As usual, only a small proportion of matrix entries are filled with an evaluation.

Instead of using similarities between lines or between columns, this exercise examines the other classical technique for recommendation : matrix factorization. In the fast few years, many methods and software tools (in python, C++, R,…) have been developed for factorizing a matrix M as the product of two matrices U and V. This method is known as Non-Negative Matrix Factorization and belongs to larger family of matrix factorization techniques, widely used in data mining (in particular text mining), signal analysis and compression.

Remarks :
- we don't usually achieve M=U.V but M is only approximately equal to U.V, the closer the better.
- the factorization technique is able to cope with empty entries in M (i.e. M is a sparse matrix). In this case, comparison of M and U.V is only done where an entry for M is available. Yet all entries of U and V are filled.
- This factorization is not something you could easily calculate manually, like a matrix product.

Write down the factorization. A new dimension appears. This new dimension is typically chosen to be much smaller than the dimensions of M (say, 10).

How can we use matrix factorization to make prediction of missing ratings in M ?
How many entries are there in U and V, how does it compare to the number of entries in M ?
How could you interpret the line-column scalar product involved in computing each entry of U.V ?
How could you interpret the new dimension introduced in the factorization process ?
How could you interpret the matrix U.transposed(U) and the matrix V.transposed(V) ?

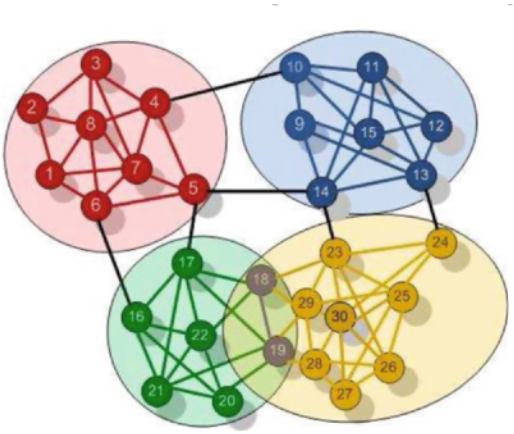
How does the concept of co-occurrence (« people who buy X generally also buy Y ») appear in matrix factorization ?

When describing the similarity between using users decomposed on topics, do we face the same spareness issue as when measuring the similarity between users in k-nearest neighbours technique ?

What is the relation between matrix factorization and multi-layer perceptron neural networks ? As a consequence, what would you expect (hope) from using neural networks ?

Factorization and social graph

This exercise addresses the graph of relations between people on a social network (undirected edges, eg people linked are symmetrical friends).



Example with 30 people and 4 communities.
[fig. by Wang et al., Data Mining Knowledge Discovery 2010]

Consider the adjacency matrix A of the graph.
Interpret the values in A.transpose(A). Because A is symmetric, it can be factorized, with approximation as previously, as B.transpose(B). The new dimension introduced is generally significantly smaller than the number of lines in A. Write down this factorization. What interpretation could you give to the lines and columns of B ? What interpretation to the approximate reconstruction of A as the product B.transpose(B) ? Can a person belong to several communities at the same time ? How Could you use use this to make link prediction in the social network (i.e. predict that two people are likely to know each other (or to have common characteristics or interest, as far as the social graph can tell, although they are currently not linked) ?