Journal of Uncertain Systems
Vol.13, No.3, pp.163-166, 2019
Online at: www.jus.org.uk

Why Matrix Factorization Works Well in Recommender Systems: A Systems-Based Explanation

Griselda Acosta[1], Manuel Hernandez[2]
Natalia Villanueva-Rosales[2], Eric Smith[3], Vladik Kreinovich[2,*]
[1] *Department of Electrical and Computer Engineering, University of Texas at El Paso*
500 W. University, El Paso, Texas 79968, USA
[2] *Department of Computer Science, University of Texas at El Paso*
500 W. University, El Paso, Texas 79968, USA
[3] *Department of Industrial, Manufacturing, and Systems Engineering*
University of Texas at El Paso, 500 W. University, El Paso, Texas 79968, USA

Received 1 July 2019; Revised 23 July 2019

Abstract

Many computer-based services use recommender systems that predict our preferences based on our degree of satisfaction with the past selections. One of the most efficient techniques making recommender systems successful is matrix factorization. While this technique works well, until now, there was no general explanation of why it works. In this paper, we provide such an explanation.

Keywords: recommender system, matrix factorization method, systems-based explanation

1 Formulation of the Problem

Recommender systems. Many computer-based services aim at making the customers happier. For example, platforms like amazon.com that help us buy things not only allow us to buy what we want, they also advise us about we may be interested in looking at.

The system comes up with this advice based on our previous pattern of purchases and on how satisfied we were with these purchases. For example, platforms like Netflix not only allow you to watch movies, they also use our previous selections to help the customer by providing advice on what other movies this particular customer will want to see.

To make such recommendations, for each customer i, the system uses the ratings r_{ij} that different customers made for different objects j. Based on the available values r_{ij} corresponding to different customers and different objects, the system estimates the customer's future ratings of different possible objects – and, based on these ratings, recommends, to each customer i, the objects j for which the estimated ratings r_{ij} are the largest. Such systems that, based on our past selections and our previous rating, try to predict our future preferences are known as *recommender systems*.

Matrix factorization. One of the most successful techniques in designing recommender systems is *matrix factorization*; see, e.g., [2, 3] and references therein. This method is based on the assumption that we can find parameters $c_{i1},\ldots,c_{in}$ characterizing the i-th customer and parameters $o_{j1},\ldots,o_{jn}$ characterizing the j-th object so that the rating r_{ij} of i-th customer on the j-th object has the form

$$r_{ij}=\sum_{k=1}^{n} c_{ik}\cdot o_{jk}. \tag{1}$$

*Corresponding author.
Emails: gvacosta@miners.utep.edu (G. Acosta), mahernandez46@miners.utep.edu (M. Hernandez), nvillanuevarosales@utep.edu (N. Villanueva-Rosales), esmith2@utep.edu(E. Smith), vladik@utep.edu (V. Kreinovich).

Challenge. While the matrix factorization methods works well, it is not clear why a person's recommendations can be described in this way.

What we do in this paper. In this paper, we provide a general systems-based explanation for the matrix factorization method.

2 Why Matrix Factorization: Our Explanation

Formulation of the problem in precise terms. Let $p_1, \ldots, p_n$ be parameters describing a customer, and let $q_1, \ldots, q_n$ be parameters describing the object. Based on the values p_i describing the customer and on the values $q_1, \ldots, q_n$ describing the object, we need to estimate the customer's rating of the object. Let us denote the algorithm providing such an estimation by

$$f(p_1, \cdots, p_n, q_1, \cdots, q_n).$$

We want to explain why the formula (1) is a good model for such a dependence.

Linearization. In general, in the first approximation, we can always expand each dependence in Taylor series and keep only linear terms in the corresponding expansion. This *linearization* procedure is a general systems idea widely (and successfully) used in physics, in engineering, and in many other applications; see, e.g., [1].

In line with this general idea, let us expand the function f in Taylor series in terms of the values $p_1, \ldots, p_n$ and keep only linear terms in this expansion. We perform this procedure for each possible combination of values $q_1, \ldots, q_n$. As a result, for each possible combination of values $q_1, \ldots, q_n$, we get an expression which is linear in p_i:

$$f(p_1, \cdots, p_n, q_1, \cdots, q_n) = a_0(q_1, \cdots, q_n) + \sum_{k=1}^{n} a_k(q_1, \cdots, q_n) \cdot p_k. \tag{2}$$

In this expression, in general, for different combinations of values q_k, the corresponding coefficients a_k are different.

We can then apply the same linearization procedure to each of the dependencies $a_k(q_1, \cdots, q_n)$:

$$a_k(q_1, \cdots, q_n) = a_{k0} + \sum_{\ell=1}^{n} a_{k\ell} \cdot q_\ell. \tag{3}$$

Substituting the expressions (3) into the formula (2), we conclude that

$$f(p_1, \cdots, p_n, q_1, \cdots, q_n) = a_{00} + \sum_{k=1}^{n} a_{k0} \cdot p_k + \sum_{\ell=1}^{n} a_{0\ell} \cdot q_\ell + \sum_{k=1}^{n} \sum_{\ell=1}^{n} a_{k\ell} \cdot p_k \cdot q_\ell. \tag{4}$$

Singular Value Decomposition. By selecting appropriate linear combinations of p_i and q_j, we can represent the matrix $a_{k\ell}$ in the diagonal form; this is known as the *Singular Value Decomposition* of the matrix $a_{k\ell}$. In other words, if:

- instead of the original variables $p_1, \ldots, p_n$, we use their appropriate linear combinations p'_k, and
- instead of the original variables $q_1, \ldots, q_n$, we use their appropriate linear combinations q'_ℓ,

then the expression $\sum\limits_{k=1}^{n} \sum\limits_{\ell=1}^{n} a_{k\ell} \cdot p_k \cdot q_\ell$ takes a diagonalized form $\sum\limits_{k=1}^{n} \lambda_k \cdot p'_k \cdot q'_k$ for some values λ_k.

This expression can be further simplified if instead of the variables p'_k, we use variables $p''_k \stackrel{\text{def}}{=} \lambda_k \cdot p'_k$. Then, the diagonalized form takes the following simpler form: $\sum\limits_{k=1}^{n} p''_k \cdot q'_k$.

Since the new variables p''_k are linear combinations of the original variables $p_1, \ldots, p_n$, vice versa, the original variables p_k are linear combinations of the new variables $p''_1, \ldots, p''_n$. If we substitute these linear combinations into the formula $\sum\limits_{k=1}^{n} a_{k0} \cdot p_k$, we get a linear combination of the new variables p''_k, i.e., an expression of the type $\sum\limits_{k=1}^{n} a''_{k0} \cdot p''_k$, for appropriate coefficients a''_{k0}.

Similarly, since the new variables q_ℓ are linear combinations of the original variables $q_1, \ldots, q_n$, vice versa, the original variables q_ℓ are linear combinations of the new variables $q'_1, \ldots, q'_n$. If we substitute these linear combinations into the formula $\sum\limits_{\ell=1}^{n} a_{0\ell} \cdot q_\ell$, we get a linear combination of the new variables q'_ℓ, i.e., an expression of the type $\sum\limits_{\ell=1}^{n} a'_{0\ell} \cdot q'_\ell$, for appropriate coefficients $a'_{0\ell}$.

Thus, in terms of the new variables p''_i and q'_j, the expression (4) takes the form

$$f(p''_1, \cdots, p''_n, q'_1, \cdots, q'_n) = a_{00} + \sum_{k=1}^{n} a''_{k0} \cdot p''_k + \sum_{\ell=1}^{n} a'_{0\ell} \cdot q'_\ell + \sum_{k=1}^{n} p''_k \cdot q'_k. \tag{5}$$

This expression can be further simplified. The above expression can be further simplified if we introduce new variables $p'''_k = p''_k + a'_{0k}$ and $q''_\ell = q'_\ell + a''_{\ell 0}$ for which $p''_k = p'''_k - a'_{0k}$ and $q'_\ell = q''_\ell - a''_{\ell 0}$. Substituting these expressions for p''_k and q'_ℓ into the formula (5), we get

$$\begin{aligned}
&f(p'''_1, \cdots, p'''_n, q''_1, \cdots, q''_n) \\
&= a_{00} + \sum_{k=1}^{n} a''_{k0} \cdot (p'''_k - a'_{0k}) + \sum_{\ell=1}^{n} a'_{0\ell} \cdot (q''_\ell - a''_{\ell 0}) + \sum_{k=1}^{n} (p'''_k - a'_{0k}) \cdot (q''_k - a''_{k0}) \\
&= a_{00} + \sum_{k=1}^{n} a''_{k0} \cdot p'''_k - \sum_{k=1}^{n} a''_{k0} \cdot a'_{0k} + \sum_{k=1}^{n} a'_{0k} \cdot q'_k - \sum_{k=1}^{n} a'_{0k} \cdot a''_{k0} + \\
&\qquad \sum_{k=1}^{n} p'''_k \cdot q''_k - \sum_{k=1}^{n} a'_{k0} \cdot p'''_k - \sum_{k=1}^{n} a'_{0k} \cdot q''_k + \sum_{k=1}^{n} a''_{k0} \cdot a'_{0k} \\
&= a'_{00} + \sum_{k=1}^{n} p'''_k \cdot q''_k,
\end{aligned}$$

where

$$a'_{00} = a_{00} - \sum_{k=1}^{n} a''_{k0} \cdot a'_{0k}.$$

Conclusion. Thus, in the new variables p'''_k and q''_k, the function

$$f(p'''_1, \cdots, p'''_n, q''_1, \cdots, q''_n)$$

that estimates the customer's ratings takes the form

$$f(p'''_1, \cdots, p'''_n, q''_1, \cdots, q''_n) = a'_{00} + \sum_{k=1}^{n} p'''_k \cdot q''_k. \tag{6}$$

This is – almost – the original expression (1), with:

- the variables $p'''_1, \ldots, p'''_n$ describing the customer and
- the variables $q''_1, \ldots, q''_n$ describing different objects.

The only difference from the formula (1) is that in our general formula (6) we have an additional constant term a'_{00}.

This term can be deleted if we appropriately re-scale the ratings, i.e., consider the ratings $r'_{ij} = r_{ij} - a'_{00}$ instead of the original ratings r_{ij}. Indeed, if, e.g., ratings on a scale from 0 to 10 satisfy the formula (1), this formula will no longer be valid if we use a different numerical scale for the same ratings – e.g., a scale from -5 to 5. In this sense, the formula (1) already pre-assumes that we are using an appropriate scale – and thus, our general formula (6) indeed provides an explanation for why the empirical formula (1) is so ubiquitous.

Acknowledgments

This work was supported in part by the National Science Foundation grants 1623190 (A Model of Change for Preparing a New Generation for Professional Practice in Computer Science) and HRD-1242122 (Cyber-ShARE Center of Excellence).

References

[1] Feynman, R., Leighton, R., and M. Sands, *The Feynman Lectures on Physics*, Addison Wesley, Boston, Massachusetts, 2005.

[2] Salakhutdinov, R., and A. Mnih, Probabilistic matrix factorization, *Advances in Neural Information Processing Systems*, pp.1257–1264, 2015.

[3] Zhao, C., Sun, S., Han, L., and Q. Peng, Hydrid matrix factorization for recommender systems in social networks, *International Journal on Neural and Mass-Parallel Computing and Information Systems*, vol.26, no.6, pp.559–569, 2016.