An Elementary View on Factorization Machines

Sebastian Prillo
Universidad de Buenos Aires
sprillo@dc.uba.ar

ABSTRACT

Factorization Machines (FMs) are a model class capable of learning pairwise (and in general higher order) feature interactions from high dimensional, sparse data. In this paper we adopt an elementary view on FMs. Specifically, we view FMs as a sum of simple surfaces - a hyperplane plus several squared hyperplanes - in the original feature space. This elementary view, although equivalent to that of low rank matrix factorization, is geometrically more intuitive and points to some interesting generalizations. Led by our intuition, we challenge our understanding of the inductive bias of FMs by showing a simple dataset where FMs *counterintuitively* fail to learn the weight of the interaction between two features. We discuss the reasons, and mathematically formulate and prove a form of this limitation. Also inspired by our elementary view, we propose modeling *intermediate orders of interaction*, such as 1.5-way FMs. Beyond the specific proposals, the goal of this paper is to expose our thoughts and ideas to the research community in an effort to take FMs to the next level.

KEYWORDS

ANOVA kernel; Factorization Machines; Machine Learning; Recommender Systems

1 INTRODUCTION

Factorization Machines (FMs), introduced by Steffen Rendle in [6] are a supervised learning model class which excels at learning pairwise feature interactions from sparse data. The origin of FMs lies in matrix factorization models commonly used in recommender system tasks. However, FMs are different in that they are not limited to specific types of data - they are a *general* predictor.

FMs are best understood by considering the following second-order polynomial model:

$$\hat{y}(x) := w_0 + \sum_{i=1}^{n} w_i x_i + \sum_{i=1}^{n} \sum_{j=i+1}^{n} w_{i,j} x_i x_j \tag{1}$$

and then demanding that the pairwise interaction weight matrix $W = (w_{i,j})_{1 \le i,j \le n}$ have low-rank representation $W = VV^T$ where $V \in \mathbb{R}^{n \times k}$ (we ignore the diagonal terms in W); k is of course typically chosen to be much smaller than n. The FM model equation thus is:

$$\hat{y}(x) := w_0 + \sum_{i=1}^{n} w_i x_i + \sum_{i=1}^{n} \sum_{j=i+1}^{n} \left\langle v_i, v_j \right\rangle x_i x_j \tag{2}$$

where $w_0 \in \mathbb{R}$, $w \in \mathbb{R}^n$ and v_i is the i-th row of $V \in \mathbb{R}^{n \times k}$.

A widely adopted interpretation of FMs is that they work by embedding each feature in a k dimensional vector space and then expressing the weight of the interaction between two features as the similarity of their vector embeddings. In [6], Rendle uses this interpretation to explain how FMs can learn user preferences from a small toy dataset of user-movie ratings.

We focus instead on the following reformulation of FMs. As Rendle has shown himself [6, Lema 3.1], the FM model equation can be rewritten as:

$$\hat{y}(x) := w_0 + \sum_{i=1}^{n} w_i x_i + \frac{1}{2} \sum_{f=1}^{k} \left(\left(\sum_{i=1}^{n} v_{i,f} x_i \right)^2 - \sum_{i=1}^{n} v_{i,f}^2 x_i^2 \right) \tag{3}$$

Rendle uses this reformulation to prove that FMs can be computed in linear time $O(nk)$ and thus efficiently trained. However, this reformulation has further meaning: it shows that FMs are (roughly) nothing more than the sum of a hyperplane plus k squared hyperplanes in the original feature space. This is our *elementary view*.

Inspired by our elementary view, we make our first contribution by showing a simple dataset where FMs counterintuitively fail to learn the weight of the interaction between two features. We discuss why this happens and dig deeper into the inductive bias of FMs, hopefully shedding some light on the way they work, and how we can use them better.

For our second contribution, we show how we might go about modeling *intermediate orders* of interaction, such as 1.5-way FMs. Several generalizations of FMs have been proposed in the past, including Higher-Order FMs (HOFM) [2, 6], Field-Aware FMs (FFMs) [5] and Convex FMs [1], however, to the best of our knowledge, intermediate-order FMs represent a novel concept.

2 ASYMMETRY OF FACTORIZATION MACHINES

2.1 Discussion

An obvious consequence of viewing FMs as the sum of a hyperplane plus k squared hyperplanes in the original feature space is that FMs exhibit some form of asymmetry with respect to the response variable. Let us make this clear by means of an example: Suppose we want to train a classifier to distinguish cats from dogs. Label 'cat' as the positive class and 'dog' as the negative class. Train a FM (say with logistic link function). Now reverse the class labels (make 'dog'

RecSys '17, August 27-31, 2017, Como, Italy

ACM ISBN 978-1-4503-4652-8/17/08…$15.00
https://doi.org/10.1145/3109859.3109892

the positive class and 'cat' the negative class) and train another FM. *Then these two FMs will perform differently*, and the reason for this does not have to do with the fact that the FM optimization problem is non-convex: it rather has to do with the fact that FMs' hypothesis space is different from its 'negative'. Formally, if FM^+ is the family of functions defined by equation 3, then by 'its negative' we mean the family $FM^- = \{-f : f \in FM^+\}$.

The fact that FMs exhibit this form of asymmetry is by no means a problem, but since it is not the typical case for a general predictor, it raises some questions regarding FMs' inductive bias. After some thought, we have come up with a simple toy dataset to show how this asymmetry might impact FMs. Consider the following dataset, which portrays the impact of several factors on life expectancy:

smokes	sedentary life	unhealthy diet	life expectancy
0	0	0	10
1	0	0	8
0	1	0	8
0	0	1	8
1	1	0	2
1	0	1	2

We purposefully omitted the observation $(0, 1, 1, 2)$ from the dataset to use it as our test case. Following Rendle's exact same line of argument as in [6, Parameter Estimation Under Sparsity], replacing 'Bob' by 'sedentary life', 'Charlie' by 'unhealthy diet', 'Star Wars' by 'smokes', and 'rating' by 'life expectancy', we get this claim: *"'sedentary life' and 'unhealthy diet' will have similar factor vectors v_2 and v_3 because both have similar interactions with 'smokes' (v_1) for predicting 'life expectancy' "*. In other words, v_2 and v_3 must be similar. But this implies that the inner product between v_2 and v_3 must be positive, and hence the pairwise weight for sedentary life and unhealthy diet must be positive. This is the opposite of what we were expecting! What went wrong?

Let us explain the previous example with concrete numbers. To make our point we will consider $k = 1$, but as we will prove, a similar result holds in higher dimensions. The FM model equation for $k = 1$ is:

$$\hat{y}(x) = w_0 + w_1x_1 + w_2x_2 + w_3x_3 + v_1v_2x_1x_2 + v_1v_3x_1x_3 + v_2v_3x_2x_3 \tag{4}$$

where $w_i, v_i \in \mathbb{R}$. Let us fit this FM to the data and see what it learns about the pairwise interaction between the features 'sedentary life' and 'unhealthy diet'; our intuition demands this weight to be negative. Note that the weights for the pairwise interactions are v_1v_2, v_1v_3 and v_2v_3, whose product is a perfect square. Thus, if the FM infers from the data that smoking and having a sedentary life should have a negative impact on life expectancy, and similarly that smoking and having an unhealthy diet should have a negative impact on life expectancy, then the FM **must** conclude that having a sedentary life and an unhealthy diet should have a positive impact on life expectancy. For example, the FM could learn the weights

$$(w_0, w_1, w_2, w_3, v_1, v_2, v_3) = (10, -2, -2, -2, -2, 2, 2)$$

which fit the data, but will surprisingly predict a life expectancy of 10 for point $(0, 1, 1)$. Note that if we reverse the effect of the independent variables on the dependent variable, say by multiplying the life expectancy by -1, then the FM can learn weights

$$(w_0, w_1, w_2, w_3, v_1, v_2, v_3) = (-10, 2, 2, 2, 2, 2, 2)$$

and correctly predict a value of -2 for point $(0, 1, 1)$.

In the general case ($k \geq 1$), it is not necessarily true that if $\langle v_1, v_2\rangle$ and $\langle v_1, v_3\rangle$ are negative, then $\langle v_2, v_3\rangle$ is positive. For larger values of k, the high-dimensionality of the embedding space allows the vectors to accommodate in more sophisticated ways. However, the following result holds:

Theorem 2.1. *Given $n + 2$ vectors in $\mathbb{R}^n$ ($n \geq 1$), there are two whose inner product is non-negative.*

Proof. We prove the proposition by induction on n. For $n = 1$ it is trivial. Suppose it is true for some $n - 1 \geq 1$ and let us prove it for n. Let $v_1, v_2, \ldots, v_{n+2}$ be $n + 2$ vectors in $\mathbb{R}^n$. If $v_1 = 0$ we are done, so suppose otherwise. If any of the other v_i ($i \neq 1$) forms a non-obtuse angle with v_1 we are also done, so suppose $\langle v_i, v_1\rangle < 0$ for all $i \neq 1$. Write $v_i = \langle v_i, v_1\rangle \frac{v_1}{|v_1|^2} + u_i$ where $u_i \in \langle v_1\rangle^\perp$. Then $u_2, u_3, \ldots, u_{n+2}$ are $n + 1$ vectors in an $n - 1$-dimensional space isomorphic to $\mathbb{R}^{n-1}$, so by the inductive hypothesis there exist $2 \leq s < t \leq n + 2$ such that $\langle u_s, u_t\rangle \geq 0$. Since $\langle v_s, v_1\rangle, \langle v_t, v_1\rangle$ are both negative it follows that $\langle v_s, v_t\rangle = \left\langle \langle v_s, v_1\rangle \frac{v_1}{|v_1|^2} + u_s, \langle v_t, v_1\rangle \frac{v_1}{|v_1|^2} + u_t \right\rangle = \frac{\langle v_s, v_1\rangle\langle v_t, v_1\rangle}{|v_1|^2} + \langle u_s, u_t\rangle > \langle u_s, u_t\rangle \geq 0$, and we are done. □

As a corollary, given a Factorization Machine with embedding of dimension k, for any $k + 2$ chosen features there will always be two whose pairwise interaction weight is non-negative. This is an easy to grasp result showing a limitation of FMs. Of course, the result is false if 'non-negative' is changed by 'non-positive', which displays the asymmetric nature of FMs.

Our analysis raises some questions regarding the inductive bias of FMs: we would like to know intuitively what criterion FMs are using to infer the pairwise interaction weights, and why this criterion is asymmetric. To answer this question, it is again convenient to look at the case $k = 1$ (larger values of k are just meant to generalize this intuition to higher dimensions) and consider a graph on n vertices, one vertex per feature. Let the edge between two vertices be assigned the weight of the pairwise interaction between those features. Assume for simplicity that all weights are non-zero. Then FMs can model precisely those graphs where:

1. For every cycle of even length, the two alternating products of edge weights are equal.
2. For every cycle of odd length, the product of its weights is positive.

From this, it is clear that FMs (with $k = 1$) have only n degrees of freedom: fix the weights of a spanning tree and an edge forming an odd cycle (in a way that does not violate condition 2), and from the above restrictions all other weights are uniquely determined. FM^- changes (2) to read 'for every cycle of odd length, the product of its weights is **negative**'. FMs 'fail' in our toy dataset because our intuition want us to infer from two negative edges in a triangle a third negative edge, but the inductive bias of FMs will do the exact opposite.

Explaining why this issue does not arise during Rendle's explanation of the toy user-movie dataset is a bit more subtle, and has to do with the fact that in this dataset the pairwise weights that are only ever exercised are induced by a bipartite graph with users on one side and movies on the other, but FM^+ and FM^- are equally expressive in this context. Formally, $FM^+ = FM^-$ when we restrict the domain of the functions in FM^+ and FM^- to the set X of vectors $x \in \{0,1\}^{U+I}$ ($U, I \in \mathbb{N}$ fixed) such that exactly one of $x_1, \ldots, x_U$ is 1, and exactly one of $x_{U+1}, \ldots, x_{U+I}$ is 1. The proof for this is easy: if $f \in FM^+$ is parametrized by the weights $(w_0, \ldots, w_n, v_1, \ldots, v_{U+I})$ as in equation 2, then $-f \in FM^+$, parametrized by the weights $(-w_0, \ldots, -w_n, -v_1, \ldots, -v_U, v_{U+1}, \ldots, v_{U+I})$, hence $FM^+ = FM^-$.

2.2 Exploiting or addressing asymmetry

The previous discussion suggests a trivial trick to try to improve the performance of FMs: since $FM^+ \neq FM^-$ (except specific cases as the one discussed), *just by reversing the response variable in the dataset* (which is equivalent to training a model from FM^- instead of FM^+) the performance of FMs might improve. Ensembling models from FM^+ and FM^- might be another good idea.

If we talk about 'fixing' the asymmetry of FMs, some good ideas can be taken from the work of [3]. Indeed, we can consider:

$$\hat{y}(x) = w_0 + \sum_{i=1}^{n} w_i x_i + \frac{1}{2}\sum_{f=1}^{k} \lambda_i \Big(\big(\sum_{i=1}^{n} v_{i,f} x_i\big)^2 - \sum_{i=1}^{n} v_{i,f}^2 x_i^2 \Big) \quad (5)$$

where $\lambda_i \in \mathbb{R}$. Since the absolute value of λ_i can be absorbed into the right hand factor, this hypothesis space as expressive as limiting each λ_i to take the value -1 or 1. From our elementary point of view, the squared hyperplanes can now be either added or subtracted. A middle ground approach which does not require fitting λ is to just force half the λ_i in equation 5 to be 1 and the other half to be -1. Although this might sound ad-hoc, it has a nice mathematical interpretation: it is the same as the original FM model equation (2) but with a $\frac{k}{2}$ (assume k even) dimensional feature embedding in $\mathbb{C}^{\frac{k}{2}}$ (hence $V \in \mathbb{C}^{n \times \frac{k}{2}}$), with the real inner product replaced by the bilinear form $\langle a + bi, c + di \rangle = \langle a, c \rangle + i \langle b, c \rangle + i \langle a, d \rangle - \langle b, d \rangle$ for $a, b, c, d \in \mathbb{R}^{\frac{k}{2}}$, and then taking the real part from the resulting equation. It is tempting to call these *complex FMs*.

Interestingly, equation 5 is closely related to the following model, also derived from [3]:

$$\hat{y}(x) := w_0 + \sum_{i=1}^{n} w_i x_i + \sum_{i=1}^{n} \sum_{j=i+1}^{n} \big\langle u_i, v_j \big\rangle x_i x_j \quad (6)$$

where $w_0 \in \mathbb{R}, w \in \mathbb{R}^n$ and $U, V \in \mathbb{R}^{n \times k}$. In essence, the asymmetry we have discussed is directly related to the positive semi-definiteness of the pairwise interaction weight matrix W. By relaxing the representation of W from $W = VV^T$ to $W = UV^T$, the positive semi-definiteness of W is bypassed. The results in [3] suggest that the models defined by equations 5 and 6 should be of comparable performance when the embedding size of the former is twice the one of the latter.

3 INTERMEDIATE-ORDER FACTORIZATION MACHINES

3.1 Motivation

The essence of FMs lies in the distributive law: when squaring a hyperplane pairwise interaction terms arise naturally, and similarly when cubing a hyperplane 3-way interaction terms arise. Of course, this also generates some unwanted terms such as $v_i^2 v_j x_i^2 x_j$ which we must deal with. But clearly, *there is nothing stopping us from exponentiating a hyperplane to an arbitrary d-th power.* By doing so, we can interpret it as giving rise to d-way interactions.

3.2 Formalization

To formalize this idea, it is convenient to speak in terms of kernels. As stated in [2, 3], the FM model equation (3) is equivalent to

$$\hat{y}(x) = w_0 + \langle w, x \rangle + \sum_{f=1}^{k} \mathcal{K}_f(c_f, x) \quad (7)$$

where $\mathcal{K}_f = \mathcal{A}_2$ are ANOVA kernels of degree 2, and c_f is the f-th column of V. Similarly, 3-way FMs are a combination of ANOVA kernels of degree 2 and 3. In general, the ANOVA kernel $\mathcal{A}_m$ of degree m is responsible for capturing interactions of order m. Thus, the goal of building intermediate-order FMs translates to that of extrapolating $\mathcal{A}_m$ to non-integral values of m. Of course, the discrete nature underlying the definition of $\mathcal{A}_m$ makes it hard to do so directly. However, by expressing $\mathcal{A}_m$ in terms of simpler quantities we can try to achieve our goal. These simpler quantities will be precisely powers of hyperplanes, as discussed in our initial motivation, among others.

Specifically, let $\mathcal{D}_m(x, y) = \sum_{i=1}^{n} x_i^m y_i^m$ be the 'power sum' kernel of degree m. That is, if e_m and p_m are respectively the elementary and power sum symmetric polynomials of degree m in n variables, and $\circ$ is the Hadamard product, then we have $\mathcal{D}_m(x, y) = p_m(x \circ y)$ and $\mathcal{A}_m(x, y) = e_m(x \circ y)$. Since the p_i $(1 \leq i \leq n)$ form a basis for the ring of symmetric polynomials over $\mathbb{Q}$ - of which e_m is such an element - then e_m admits a unique representation as a polynomial in the p_i, and in turn $\mathcal{A}_m$ admits a unique representation as polynomial in the $\mathcal{D}_i$. For instance, $\mathcal{A}_2 = \frac{1}{2}(\mathcal{D}_1^2 - \mathcal{D}_2)$ and $\mathcal{A}_3 = \frac{1}{6}(\mathcal{D}_1^3 - 3\mathcal{D}_2\mathcal{D}_1 + 2\mathcal{D}_3)$. These identities are derived in [3, Lemma 3], but the connection with the ring of symmetric polynomials was not identified. Thus, in general, for all m we have an essentially unique representation

$$\mathcal{A}_m = \sum_{i=1}^{s} q_i \prod_{j=1}^{t_i} \mathcal{D}_{m_{i,j}} \quad (8)$$

where $s \in \mathbb{N}_0, q_i \in \mathbb{Q}, t_i \in \mathbb{N}, m_{i,j} \leq n$. With this, let us define the extrapolation of $\mathcal{A}_m$ to order $d \in \mathbb{R}$ by

$$\mathcal{A}_{m \to d} = \sum_{i=1}^{s} q_i \prod_{j=1}^{t_i} \operatorname{sign}(\mathcal{D}_{m_{i,j}}) \big|\mathcal{D}_{m_{i,j}}\big|^{\frac{d}{m}} \quad (9)$$

Note that this extrapolation is consistent in the sense that $\mathcal{A}_m = \mathcal{A}_{m \to m}$ for all $m \in \mathbb{N}$, and, furthermore, $\mathcal{A}_{m \to d}$ is homogeneous of degree d, meaning $\mathcal{A}_{m \to d}(\lambda x, y) = \operatorname{sign}(\lambda^m)|\lambda|^d \mathcal{A}_{m \to d}(x, y)$. Please note that our extrapolation is by no means the only nor the best way to do this. We believe that there might be a better way to extrapolate $\mathcal{A}_m$ to non-integral values of m, perhaps in a way that seamlessly unifies all the ANOVA kernels of different degrees. We think an affirmative answer to this question would be a major breakthrough in our understanding of Factorization Machines.

3.3 Extrapolating 2-way FMs

When $m = 2$ we have $\mathcal{A}_m = \frac{1}{2}\mathcal{D}_1\mathcal{D}_1 - \frac{1}{2}\mathcal{D}_2$ and hence equation 9 leads to $\mathcal{A}_{2\to d} = \frac{1}{2}(|\mathcal{D}_1|^d - \mathcal{D}_2^{\frac{d}{2}})$. Thus, we can extrapolate 2-way FMs to d-way FMs by considering

$$\hat{y}(x) := w_0 + \sum_{i=1}^{n} w_i x_i + \frac{1}{2}\sum_{f=1}^{k}\left(\left|\sum_{i=1}^{n} v_{i,f}x_i\right|^d - \left(\sum_{i=1}^{n} v_{i,f}^2 x_i^2\right)^{\frac{d}{2}}\right) \quad (10)$$

Let us extend our previous definitions and denote FM_d^+ the family of functions defined by equation 10, and $FM_d^- = \{-f : f \in FM_d^+\}$.

3.4 Implementation

Adapting 2-way FMs to support equation 10 is easy in the case of SGD learning. We believe it might not be possible to do so efficiently in the case of ALS and MCMC learning, because the key property of multi-linearity, which leads to multi-convexity, is lost. See [7] for a full exposition on these learning procedures applied to FMs.

Perhaps surprisingly, when $d \geq 1$, equation 10 has continuous partial derivatives with respect to the $v_{i,f}$, given by:

$$\frac{\partial}{\partial v_{i,f}}\hat{y}(x) = \frac{d}{2}x_i \operatorname{sign}(D_1)|D_1|^{d-1} - \frac{d}{2}v_{i,f}x_i^2 D_2^{\frac{d-2}{2}}$$

where $D_1 = \mathcal{D}_1(c_f, x)$ and $D_2 = \mathcal{D}_2(c_f, x)$. This is similar to the derivatives of 2-way FMs and can also be computed efficiently with minor modifications to the code. The only caveat is that exponentiating a floating point number to an arbitrary exponent is *a priori* a slow operation, but this can be overcome by restricting ourselves to suitable values of d, such as $d \in \{1.25, 1.50, 1.75, \ldots\}$ for which exponentiation can be efficiently computed by taking only square roots. Note also that equation 10 can be easily differentiated with respect to d, allowing us to learn the interaction level from the data if we wish.

4 RESULTS

We modified the libFM implementation [7] to support intermediate-order FMs as defined by equation 10. We ran experiments on the Movielens 100k dataset[1] [4]; the dataset contains 10^5 ratings from 943 users on 1682 movies. We trained d-way FMs for $d \in \{1.25, 1.5, 1.75, 2\}$. We set $k = 4$, and after some experimentation settled on a learning rate of 0.01, regularization of 0.075 and initialization standard deviation of 10^{-5}, which seem to work well for these models. For each model, given a random number seed, we trained it on each of the five 80/20 c.v. splits `u*.base,u*.test` provided by the owners of the dataset. We trained for 200 iterations (enough for the models to overfit) and considered the best validation RMSE obtained during these 200 iterations. We then averaged this validation RMSE over the 5 folds. Finally, for this statistic we plotted the mean and a confidence interval corresponding to two standard deviations, computed from 30 runs with different seeds: $1 \leq seed \leq 30$. This whole experiment was run twice, the second time by first multiplying the response variable by -1.

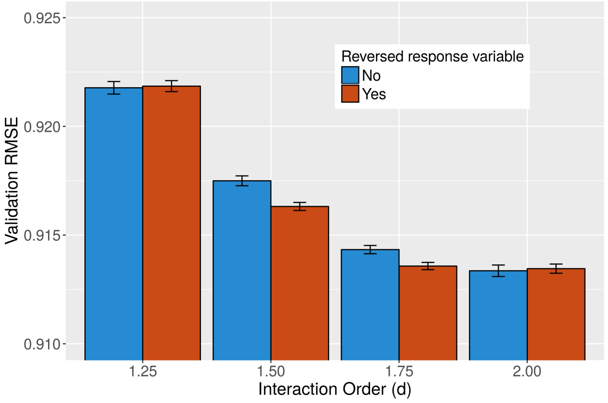

Figure 1: Effect of interaction order and reversal of the response variable on RMSE.

Several of the observations we made in this paper are reflected in the results. First of all, since the dataset consists of only two categorical variables, we have $FM_2^+ = FM_2^-$, as we have proven, and this is supported by the confidence intervals for $d = 2$ which considerably overlap. On the other hand, it is not necessarily true that $FM_d^+ = FM_d^-$ for $d \neq 2$ (our proof fails in this case), and indeed when reversing the response variable, our 1.5 and 1.75-way FMs see a boost in RMSE. Finally, we can see that 1.75-way FMs are competitive with 2-way FMs on this dataset. It is worthwhile to mention that we trained linear models which achieved an RMSE of 0.942, so even 1.25-way FMs perform significantly better than plain linear models.

We tried training these d-way FMs for $2 < d < 3$ but surprisingly, as soon as $d > 2.1$, SGD gets stuck in what appears to be a plateau with high training RMSE, only ever escaping. This seems to align with another behavior we observed, where for $1 < d < 2$, the smaller the value of d, the faster the progress SGD makes in the first iterations: it takes 2-way FMs roughly 40 iterations to reach the same RMSE as 1.25-way FMs in 10 iterations. It looks like smaller values of d might make the optimization problem more amenable for SGD by altering the curvature of the error surface, but this will require further investigation.

5 CONCLUSION

In this paper we explored an elementary view on FMs. This gave us immediate insights: we explored the asymmetry of FMs, which allowed us to better understand their inductive bias, and we proposed the modeling of intermediate orders of interaction, an abstract concept. We showed preliminary results supporting the value of our ideas. However, our work leaves several open questions: for instance, our way of modeling intermediate-order interactions feels perfectible. Our hope is that these ideas can be taken by the research community and ripened to evolve FMs to the next level.

[1] https://grouplens.org/datasets/movielens/100k/

REFERENCES

[1] Mathieu Blondel, Akinori Fujino, and Naonori Ueda. 2015. Convex Factorization Machines. In *Proceedings, Part II, of the European Conference on Machine Learning and Knowledge Discovery in Databases - Volume 9285 (ECML PKDD 2015)*. Springer-Verlag New York, Inc., New York, NY, USA, 19–35. https://doi.org/10.1007/978-3-319-23525-7_2

[2] Mathieu Blondel, Akinori Fujino, Naonori Ueda, and Masakazu Ishihata. 2016. Higher-Order Factorization Machines. In *Advances in Neural Information Processing Systems 29*, D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett (Eds.). Curran Associates, Inc., 3351–3359. http://papers.nips.cc/paper/6144-higher-order-factorization-machines.pdf

[3] Mathieu Blondel, Masakazu Ishihata, Akinori Fujino, and Naonori Ueda. 2016. Polynomial Networks and Factorization Machines: New Insights and Efficient Training Algorithms. In *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48 (ICML'16)*. JMLR.org, 850–858. http://dl.acm.org/citation.cfm?id=3045390.3045481

[4] F. Maxwell Harper and Joseph A. Konstan. 2015. The MovieLens Datasets: History and Context. *ACM Trans. Interact. Intell. Syst.* 5, 4, Article 19 (Dec. 2015), 19 pages. https://doi.org/10.1145/2827872

[5] Yuchin Juan, Yong Zhuang, Wei-Sheng Chin, and Chih-Jen Lin. 2016. Field-aware Factorization Machines for CTR Prediction. In *Proceedings of the 10th ACM Conference on Recommender Systems*.

[6] Steffen Rendle. 2010. Factorization Machines. In *Proceedings of the 2010 IEEE International Conference on Data Mining*.

[7] Steffen Rendle. 2012. Factorization Machines with libFM. *ACM Trans. Intell. Syst. Technol.* 3 (2012).