# [Datawhale Paper Share] Field-matrixed Factorization Machines for Recommender Systems. WWW, 2021.

Enneng Yang

Northeastern University

18th April 2021

# $FM^2$: Field-matrixed Factorization Machines for Recommender Systems

Yang Sun
Yahoo Research
Sunnyvale, CA, USA
yang.sun@verizonmedia.com

Junwei Pan
Yahoo Research
Sunnyvale, CA, USA
pandevirus@gmail.com

Alex Zhang
Yahoo Research
Sunnyvale, CA, USA
alex.zhang@verizonmedia.com

Aaron Flores
Yahoo Research
Sunnyvale, CA, USA
aaron.flores@verizonmedia.com

# CTR: example

An example of multi-field categorical data for CTR prediction.

| CLICK | User_ID | GENDER | ADVERTISER | PUBLISHER |
|---|---|---|---|---|
| 1 | 29127394 | Male | Nike | news.yahoo.com |
| -1 | 89283132 | Female | Walmart | techcrunch.com |
| -1 | 91213212 | Male | Gucci | nba.com |
| -1 | 71620391 | Female | Uber | tripadviser.com |
| 1 | 39102740 | Male | Adidas | mlb.com |

## CTR: Feature[1]

Sparse Features:

- Number of features: $\sim$ Million to Billion
- Number of active features per sample/request: $\sim$ Ten to a Thousand

Field:

- examples:
  - Advertisers: Nike, Adidas, ...
  - AdvertisementID: 1234, 3456, ...
  - Gender: male, female, unknown
- Number of Field: $\sim$ Ten to a Thousand
- Each feature belongs to a field

[1] Referred to junwei pan's slide.

## CTR: feature embedding

Embedding: raw sparse features $\rightarrow$ dense vectors.

- raw sparse features

$$\mathbf{X} = [\mathbf{X}_1; \mathbf{X}_2; \ldots; \mathbf{X}_\mathbf{N}] \tag{1}$$

  - $\mathbf{N}$: the number of total feature fields
  - $\mathbf{X_i}$: the feature representation (one-hot vector in usual) of the i-th field

- embedding vector $\mathbf{v_i}$

$$\begin{aligned} \mathbf{v_i} &= \mathbf{V_i X_i}, \text{ where } \mathbf{V_i} \in \mathbf{R}^{n_i \times d} \\ \mathbf{V} &= \{\mathbf{V_1}, \mathbf{V_2}, \ldots, \mathbf{V_N}\} \end{aligned} \tag{2}$$

  - $n_i$: the number of features in the i-th field
  - $d$: the size for of embedding vectors

## CTR: Object & Training

- prediction score

$$\hat{y} = \phi(\mathbf{X} \mid \mathbf{V}, \Theta) \tag{3}$$

  - $\Theta$: model' s other parameters
  - $\phi()$: FM, DeepFM, xDeepFM, AutoInt ...

- training loss

$$\min \mathcal{L}(\mathbf{V}, \Theta, \mathcal{D}) \tag{4}$$

  - $\mathcal{D} = \{\mathbf{X}, y\}$ represent the training data fed into the model
  - $\mathcal{L}$ is the Logloss:

$$\mathcal{L} = -\frac{1}{|\mathcal{D}|} \sum_{j=1}^{|\mathcal{D}|} \left(y_j \log\left(\hat{y}_j\right) + \left(1 - y_j\right) \log\left(1 - \hat{y}_j\right)\right) \tag{5}$$

## CTR: Related Works Overview

CTR Models:

- shallow: LR, Poly2, FM, FFM, FwFM, FvFM, FmFM
- deep: FNN, PNN, Wide&Deep, DeepFM, Deep&Cross, xDeepFM, AutoInt, AFN ...

# CTR: Related Works Overview

**Logistic Regression (LR)**: a linear combination of individual features.

$$\Phi_{LR}(\boldsymbol{w}, \boldsymbol{x}) = w_0 + \sum_{i=1}^{m} w_i x_i \tag{6}$$

However, linear models lack the capability to represent the feature interactions.

## CTR: Related Works Overview

**Degree-2 Polynomial (Poly2)**: can effectively capture the effect of feature interactions.

$$\Phi_{\text{Poly 2}}(\boldsymbol{w}, \boldsymbol{x}) = w_0 + \sum_{i=1}^{m} w_i x_i + \sum_{i=1}^{m} \sum_{j=i+1}^{m} w_{(i,j)} x_i x_j \tag{7}$$

However, number of parameters in the model would be in the order $O(m^2)$.

## CTR: Related Works Overview

**Factorization Machines (FM)**: FM model the interaction between two features $i$ and $j$ as the dot product of their corresponding embedding vectors $v_i, v_j$.

$$\Phi_{FM}((\boldsymbol{w}, \boldsymbol{v}), \boldsymbol{x}) = w_0 + \sum_{i=1}^{m} w_i x_i + \sum_{i=1}^{m} \sum_{j=i+1}^{m} \langle \boldsymbol{v}_i, \boldsymbol{v}_j \rangle x_i x_j \tag{8}$$

However, FM neglect the fact that a feature might behave differently when it interacts with features from different other fields.

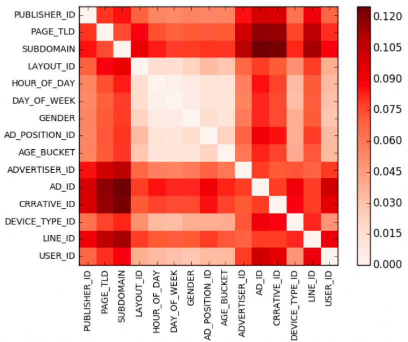
## CTR: Field Strength

To validate the heterogeneity of the field pair interactions, we use mutual information[2] between a field pair $(F_k, F_l)$ and label variable $Y$ to quantify the interaction strength of the field pair [PXR+18]:

$$MI\left((F_k, F_l), Y\right) = \sum_{(i,j)\in(F_k,F_l)} \sum_{y\in Y} p((i,j),y) \log \frac{p((i,j),y)}{p(i,j)p(y)} \tag{9}$$

[2] Mutual Information: https://en.wikipedia.org/wiki/Mutual_information

# CTR: Field Strength




Unsurprisingly, the interaction strengths of different field pairs are quite different. Some field pairs have very strong interactions, such as (AD_ID, SUBDOMAIN), (CREATIVE_ID, PAGE_TLD) while some other field pairs have very weak interactions, such as (LAYOUT_ID, GENDER), (DAY_OF_WEEK, AD$_P$*OSITION_ID*).

## CTR: Related Works Overview

**Field-aware Factorization Machines (FFM)**: model such difference explicitly by learning $n-1$ embedding vectors for each feature.

$$\Phi_{FFM}((\boldsymbol{w}, \boldsymbol{v}), \boldsymbol{x}) = w_0 + \sum_{i=1}^{m} w_i x_i + \sum_{i=1}^{m} \sum_{j=i+1}^{m} \left\langle \boldsymbol{v}_{i,F(j)}, \boldsymbol{v}_{j,F(i)} \right\rangle x_i x_j \tag{10}$$

However, their number of parameters is in the order of $O(mnk)$.

# CTR: Related Works Overview

**Field-weighted Factorization Machines (FwFM)**: models the different field interaction strength explicitly.

$$\Phi_{FwFM}((w,v),x) = w_0 + \sum_{i=1}^{m} x_i w_i + \sum_{i=1}^{m} \sum_{j=i+1}^{m} x_i x_j \langle v_i, v_j \rangle r_{F(i),F(j)} \tag{11}$$

However, FwFM has only 1 degree of freedom. (FM$=0$).

## Field-matrixed FM

**Field-matrixed Factorization Machines (FmFM)**[SPZF21]:
FmFM are extensions of FwFM in that it uses a 2-dimensional matrix instead of a scalar.

$$\Phi_{FmFM}((w,v),x) = w_0 + \sum_{i=1}^{m} x_i w_i + \sum_{i=1}^{m}\sum_{j=i+1}^{m} x_i x_j \left\langle v_i M_{F(i),F(j)}, v_j \right\rangle \tag{12}$$

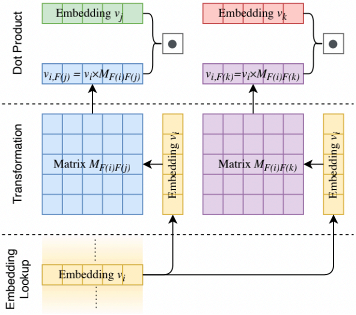
**Field-Embedded Factorization Machines (FEFM)**[Pan20]

$$\Phi_{FEFM}((w,v,W),x) = w_0 + \sum_{i=1}^{m} w_i x_i + \sum_{i=1}^{m}\sum_{j=i+1}^{m} v_i^T W_{F(i),F(j)} v_j x_i x_j \tag{13}$$

# The United Framework of Factorization Machines' Family

**Field-matrixed Factorization Machines (FmFM)**[SPZF21]:

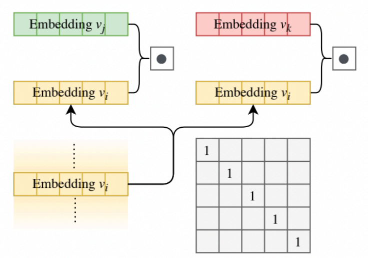
$$\Phi_{FmFM}((w,v),x) = w_0 + \sum_{i=1}^{m} x_i w_i + \sum_{i=1}^{m} \sum_{j=i+1}^{m} x_i x_j \left\langle v_i M_{F(i),F(j)}, v_j \right\rangle \tag{14}$$



# The United Framework of Factorization Machines' Family

**FM**:

$$v_i = v_i I_K \tag{15}$$



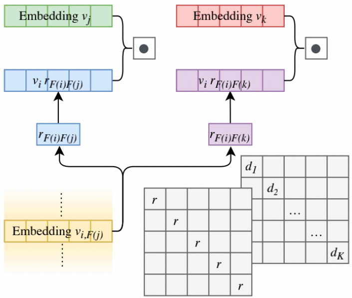
## The United Framework of Factorization Machines' Family

**FwFM**:

$$v_{i,F(j)} = v_i r_{F(i)F(j)} = v_i \left( r_{F(i)F(j)} I_K \right) \tag{16}$$

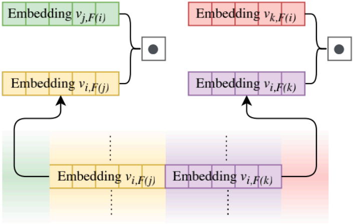
**FvFM (Field-vectorized Factorization Machines)**

$$v_{i,F(j)} = v_i D_{F(i)F(j)} = v_i \odot d_{F(i)F(j)} \tag{17}$$



# The United Framework of Factorization Machines' Family

**FFM**:



# The United Framework of Factorization Machines' Family

**FmFM v.s. OPNN**:

$$\Phi_{FmFM}((\boldsymbol{w}, \boldsymbol{v}), \boldsymbol{x}) = w_0 + \sum_{i=1}^{m} x_i w_i + \sum_{i=1}^{m} \sum_{j=i+1}^{m} x_i x_j p\left(v_i, \boldsymbol{v}_j, \boldsymbol{W}_{F(i),F(j)}\right)$$

where $\boldsymbol{W}_{F(i),F(j)} \in \mathbb{R}^{K \times K}$, and

$$p\left(v_i, v_j, W_{F(i),F(j)}\right) = \sum_{k=1}^{K} \sum_{k'=1}^{K} v_i^k v_j^{k'} w_{F(i),F(j)}^{k,k'}$$

- OPNN: outer product
- FmFM: weighted outer product

# The United Framework of Factorization Machines' Family

## Model Freedom (FM v.s. FwFM v.s. FmFM):

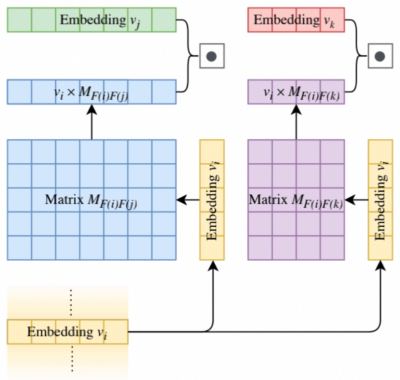
| Model | Field Interaction | Degree of Freedom |
|---|---|---|
| FM | Constant | 0 |
| FwFM | Scalar | 1 |
| FvFM | Vector | 2 |
| FmFM | Matrix | 3 |

## Model Complexity:

| Model | N of Parameters | Estimated N in Criteo Dataset |
|---|---|---|
| LR | $m$ | 1.33M |
| Poly2 | $m + H$ | 45M |
| FM | $m + mK$ | 14.63M |
| FwFM | $m + mK + \frac{n(n-1)}{2}$ | 14.63M |
| FmFM | $m + mK + \frac{n(n-1)}{2}K^2$ | 14.63M |
| FFM | $m + m(n-1)K$ | 859.18M |

# Variable Dimensions in Embeddings

$$\left\langle v_i M_{F(i)F(j)}, v_j \right\rangle = \left\langle v_j M_{F(i)F(j)}^T, v_i \right\rangle \tag{18}$$



# FmFM: Variable Dimensions in Embeddings

To optimize the field-specific embedding vector dimension without model performance loss, we propose a 2-pass method.

- In the first pass, we use a larger fixed embedding vector dimension for all fields, e.g. $k = 16$, and train the FmFM as a full model.
- From the full model, we learn how much information (variance) in each field, then we utilize a standard PCA dimensionality reduction to the embedding table in each field.

From the experiment we found that the new dimension which contains $95\%$ original variance is a good trade-off.

# FmFM: Variable Dimensions in Embeddings

standard PCA dimensionality reduction[3]: $\mathbf{X} \in \mathbb{R}^{k \times n}$

1. Zero mean: $\bar{\mathbf{X}}$
2. Covariance matrix: $C = \frac{1}{k} X X^\top$
3. The eigenvalues $\lambda_1, \ldots, \lambda_k$ and corresponding eigenvectors $p_1, \ldots, p_k$ of the C.
4. Arrange the $p_i$ according to the size of corresponding $\lambda_i$ from top to bottom.
5. Information proportion (variance)[4]: $r = \sqrt{\frac{\sum_{i=1}^{k'} \lambda_i^2}{\sum_{i=1}^{k} \lambda_i^2}}$
6. New field embedding: $X' = P'X$

[3] https://zhuanlan.zhihu.com/p/77151308

[4] https://blog.csdn.net/yly_3026925713/article/details/105058392

# FmFM: Variable Dimensions in Embeddings

| Feature Field ID | Emb Dim | Feat. N in Field | Feature Field ID | Emb Dim | Feat. N in Field |
|---|---|---|---|---|---|
| Field #01 | 3 | 62 | Field #21 | 8 | 633 |
| Field #02 | 8 | 113 | Field #22 | 2 | 3 |
| Field #03 | 5 | 125 | Field #23 | 13 | 46,329 |
| Field #04 | 7 | 50 | Field #24 | 14 | 5,228 |
| Field #05 | 9 | 223 | Field #25 | 8 | 243,452 |
| Field #06 | 8 | 147 | Field #26 | 13 | 3,176 |
| Field #07 | 6 | 99 | Field #27 | 4 | 26 |
| Field #08 | 5 | 78 | Field #28 | 14 | 11,744 |
| Field #09 | 8 | 103 | Field #29 | 10 | 225,320 |
| Field #10 | 3 | 8 | Field #30 | 6 | 10 |
| Field #11 | 5 | 31 | Field #31 | 14 | 4,726 |
| Field #12 | 3 | 56 | Field #32 | 12 | 2,056 |
| Field #13 | 6 | 81 | Field #33 | 2 | 3 |
| Field #14 | 8 | 1,457 | Field #34 | 9 | 238,638 |
| Field #15 | 12 | 555 | Field #35 | 4 | 16 |
| Field #16 | 2 | 245,195 | Field #36 | 6 | 15 |
| Field #17 | 11 | 166,164 | Field #37 | 12 | 67,854 |
| Field #18 | 5 | 305 | Field #38 | 7 | 87 |
| Field #19 | 4 | 18 | Field #39 | 11 | 50,940 |
| Field #20 | 14 | 12,054 | | | |

For example, the embedding table of field user_gender may only need 5-dimension (5D), while the field top_domain may need 7D. The field matrix M should be set up with a shape in (7, 5).

# Experiment

Statistics of training, validation and test sets of the Criteo data sets.

| Data set | | Samples | Fields | Features | Pos:Neg |
|---|---|---|---|---|---|
| Criteo | Train | 36,672,493 | 39 | 1,327,180 | ~1:3 |
| | Validation | 4,584,062 | | | |
| | Test | 4,584,062 | | | |
| Avazu | Train | 32,343,173 | 23 | 1,544,257 | ~1:5 |
| | Validation | 4,042,897 | | | |
| | Test | 4,042,897 | | | |

## Experiment

Comparison among models on Avazu CTR data sets.

| Models | AUC | | | Log Loss |
|---|---|---|---|---|
| | Training | Validation | Test | (Test Set) |
| LR | 0.7526 | 0.7521 | 0.7517 | 0.3953 |
| FM | 0.7744 | 0.7696 | 0.7695 | 0.3857 |
| FFM | **0.8012** | 0.7761 | 0.7761 | 0.3826 |
| FwFM | 0.7822 | 0.7730 | 0.7731 | 0.3835 |
| FvFM(ours) | 0.7836 | 0.7732 | 0.7733 | 0.3834 |
| FmFM(ours) | 0.7943 | **0.7764** | **0.7763** | **0.3822** |
| Deep & Cross | 0.8109 | 0.7825 | 0.7826 | 0.3791 |
| AutoInt | - | - | 0.7752 | 0.3823 |
| Fi-GNN | - | - | 0.7762 | 0.3825 |
| FGCNN+IPNN | - | - | 0.7883 | 0.3746 |
| DeepLight | - | - | **0.7897** | **0.3748** |

# Experiment

Comparison among models on the Criteo CTR data sets.

| Models | AUC | | | Log Loss |
|---|---|---|---|---|
| | Training | Validation | Test | (Test Set) |
| LR | 0.7930 | 0.7918 | 0.7917 | 0.4582 |
| FM | 0.8142 | 0.8075 | 0.8075 | 0.4441 |
| FFM | **0.8230** | 0.8103 | 0.8103 | 0.4414 |
| FwFM | 0.8191 | 0.8092 | 0.8092 | 0.4426 |
| FvFM(ours) | 0.8192 | 0.8102 | 0.8101 | 0.4417 |
| FmFM(ours) | 0.8183 | **0.8109** | **0.8109** | **0.4410** |
| Deep & Cross | 0.8244 | 0.8118 | 0.8118 | 0.4413 |
| Wide & Deep | - | - | 0.7981 | 0.4677 |
| DeepFM | - | - | 0.8007 | 0.4508 |
| xDeepFM | - | - | 0.8052 | 0.4418 |
| AutoInt | - | - | 0.8061 | 0.4454 |
| FiBiNET | - | - | 0.8103 | 0.4423 |
| DeepLight | - | - | **0.8123** | **0.4395** |

## Experiment

Compare among FmFM optimized models with embedding dim optimization, an example of the Criteo Data Set.

| Variance % | Emb Dim (Average) | FLOPs Estimated # | AUC (Test Set) | Log Loss (Test Set) |
|---|---|---|---|---|
| Full | 16(100%) | 24,531(100%) | 0.8109 | 0.4410 |
| 99% | 10.56(66.0%) | 12,884(52.5%) | 0.8109 | 0.4410 |
| 97% | 8.69(54.3%) | 10,280(41.9%) | 0.8107 | 0.4411 |
| **95%** | **7.72(48.2%)** | **8,960(36.5%)** | **0.8108** | **0.4411** |
| 90% | 6.26(39.1%) | 7,202(29.4%) | 0.8103 | 0.4415 |
| 85% | 3.82(23.9%) | 4,716(19.2%) | 0.8084 | 0.4432 |
| 80% | 3.36(21.0%) | 4,392(17.9%) | 0.8080 | 0.4436 |

# Code & Source

- **Paper**: https://arxiv.org/pdf/2102.12994.pdf
- **Code**: https://github.com/VerizonMedia/FmFM
- **Blog**: https://mp.weixin.qq.com/s/6x2VKkAlRBEm5xFVCYInEg
- **DataSet**:
  1. Avazu: https://www.kaggle.com/c/avazu-ctr-prediction/data
  2. Criteo: http://labs.criteo.com/2014/02/kaggle-display-advertising-challenge-dataset/

1. Background
2. Field-matrixed FM
3. United Framework
4. Code & Source
5. References

[Pan20] Harshit Pande.
Field-embedded factorization machines for click-through rate prediction.
*CoRR*, abs/2009.09931, 2020.

[PXR+18] Junwei Pan, Jian Xu, Alfonso Lobos Ruiz, Wenliang Zhao, Shengjun Pan, Yu Sun, and Quan Lu.
Field-weighted factorization machines for click-through rate prediction in display advertising.
In *WWW*, pages 1349–1357. ACM, 2018.

[SPZF21] Yang Sun, Junwei Pan, Alex Zhang, and Aaron Flores.
Field-matrixed factorization machines for recommender systems.
In *WWW*. ACM, 2021.

# Thanks for Listening![5]

✉contact me: ennengyang@qq.com

[5]Note: The content of this slide is for study only.