1 Factorization Machines

1.1 Introduction

Different models can be used for machine learning accordind to the task to be performed. Supervised learning is one of the tasks, where a function is learned that associates an input to an output based on example input-output pairs. Support Vector Machines (SVMs) are a set of supervised learning techniques designed to solve discrimination and regression problems. They are one of the most popular predictors in Machine Learning. However in some cases, SVMs are not efficient, because they can't learn reliable parameters in non-linear kernel spaces under very sparse data (ie data where only a few variables for each input vector of predictor variables are non-zero). Thus, the best models are either direct applications of standard or specialized factorization models. However, these models also have the disadvantage that they are not applicable to standard prediction data.
In this report, a new supervised machine learning model, called Factorization Machines (FMs) will be introduced. It's a new class of models that combines the advantages of SVMs and factorization models. FMs are especially popular in predicting the click-through rate (CTR) or for recommender systems. FMs are an impactful model and have shown excellent prediction capabilities. They have many advantages:

- FMs are a general predictor that can work with any real valued feature vector.
- FMs model all interactions between variables using factorized parameters.
- FMs estimate interactions in problems with a high degree of sparsity.
- FMs have linear complexity, can be learned in the primal and don't depend of support vectors like SVMs.

This report will contain 5 main points:

- FMs and their properties.
- Mathematical modeling of FMs.
- Comparison to other models.
- Use-cases of FMs
- Limits of FMs.

1.2 Factorization Machine

1.2.1 Factorization Machine Model

1.2.1.1 Model Equation

The most common prediction task is to estimate a function $y : \mathbb{R}^n \rightarrow T$ from a real valued feature vector $x \in \mathbb{R}^n$ to a target domain T. Let $m(x)$ be the number of non-zero elements in the feature vector x and $\bar{m}_D$ be the average number of non-zero elements $m(x)$ of all vectors $x \in D$, where D=$(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), ...$ is the training dataset.
For a factorization machine of degree d = 2 (also called a 2-way FM), the model equation is defined as:

$$\hat{y}(x) = w_0 + \sum_{i=1}^{n} w_i x_i + \sum_{i=1}^{n} \sum_{j=i+1}^{n} < v_i, v_j > x_i x_j \tag{1.1}$$

where the model parameters that have to be estimated are:

$$w_0 \in \mathbb{R}, \mathbf{w} \in \mathbb{R}^{\mathbf{n}}, \mathbf{V} \in \mathbb{R}^{\mathbf{nxk}} \tag{1.2}$$

With

$$< v_i, v_j >= \sum_{f=1}^{k} v_{i,f} v_{j,f} \tag{1.3}$$

A row $\mathbf{v_i}$ within $\mathbf{V}$ describes the i-th variable with k factors, $k \in \mathbb{N}_0^+$ is a hyperparameter that defines the dimensionality of the factorization.
A 2-way FM captures all single and pairwise interactions between variables:

- w_0 is the global bias.
- w_i models the strength of the i-th variable.
- $\hat{w}_{i,j} =< v_i, v_j >$ models the interaction between the ith and j-th variable.

The FM models the interaction by factorizing it instead of using an own model parameter $w_{i,j} \in \mathbb{R}$ for each interaction.

1.2.1.2 Expressiveness

We know that if a matrix W is positive definite , there exists a matrix V such that $W = VV^t$ provided that k is sufficiently large, under this condition, a FM can express any interaction matrix W. However in sparse settings, when there is not enough data to estimate complex interactions W, one should chose a small k. In this case, restricting k (and thus the expressiveness of the FM) improves interaction matrices under sparsity.

1.2.1.3 Parameter Estimation Under Sparsity

Factorization Machines can estimate directly and independently the interactions between variables even in sparse settings here there is not enough data. This is possible by breaking the independence of the interaction parameters by factorizing them.

1.2.1.4 Computation

The complexity of straight forward computation of eq.(1.1) is in $O(kn^2)$ because all pairwise interactions have to be computed. By reformulating pairwise interactions, it drops to linear runtime as follows :

$$\begin{aligned}
&\sum_{i=1}^{n}\sum_{j=i+1}^{n}\langle \mathbf{v}_i, \mathbf{v}_j\rangle\, x_i\, x_j \\
=&\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\langle \mathbf{v}_i, \mathbf{v}_j\rangle\, x_i\, x_j - \frac{1}{2}\sum_{i=1}^{n}\langle \mathbf{v}_i, \mathbf{v}_i\rangle\, x_i\, x_i \\
=&\frac{1}{2}\left(\sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{f=1}^{k} v_{i,f}\, v_{j,f}\, x_i\, x_j - \sum_{i=1}^{n}\sum_{f=1}^{k} v_{i,f}\, v_{i,f}\, x_i\, x_i\right) \\
=&\frac{1}{2}\sum_{f=1}^{k}\left(\left(\sum_{i=1}^{n} v_{i,f}\, x_i\right)\left(\sum_{j=1}^{n} v_{j,f}\, x_j\right) - \sum_{i=1}^{n} v_{i,f}^2\, x_i^2\right) \\
=&\frac{1}{2}\sum_{f=1}^{k}\left(\left(\sum_{i=1}^{n} v_{i,f}\, x_i\right)^2 - \sum_{i=1}^{n} v_{i,f}^2\, x_i^2\right)
\end{aligned}$$

This equation has only linear complexity, its computation is in $O(kn)$. In the case of sparsity, most of the elements in x are 0, so the sums have only to be computed over the non-zero elements. Thus, the computation of the factorization machine is in $O(k\bar{m}_D)$.

1.2.2 Factorization Machines as Predictors

FM can be applied to a variety of prediction tasks :

- Regression
- Binary classification
- Ranking

In order to prevent over-fitting, regularization terms like $L2$ are usually added to optimization objective.

1.2.3 Learning Factorization Machines

As said before, FMs have a model equation with a linear complexity. Thus, the model parameters (w_0, w, V) can be learned efficiently by gradient descent methods for a variety of losses, among them are square, logit or hinge loss.

The gradient of the FM model is calculated as follows:

$$\frac{\partial}{\partial\theta}\hat{y}(\mathbf{x}) = \begin{cases} 1, & \text{if } \theta \text{ is } w_0 \\ x_i, & \text{if } \theta \text{ is } w_i \\ x_i \sum_{j=1}^{n} v_{j,f} x_j - v_{i,f} x_i^2, & \text{if } \theta \text{ is } v_{i,f} \end{cases} \qquad (4)$$

$\sum_{j=1}^{n} v_{j,f} x_j$ can be precomputed since it's independent of i .

1.2.4 d-way Factorization Machine

We can easily generalize the 2-way FM to a d-way FM :

$$\begin{aligned}\hat{y}(x) &:= w_0 + \sum_{i=1}^{n} w_i\, x_i \\ &+ \sum_{l=2}^{d} \sum_{i_1=1}^{n} \cdots \sum_{i_l=i_{l-1}+1}^{n} \left(\prod_{j=1}^{l} x_{i_j} \right) \left(\sum_{f=1}^{k_l} \prod_{j=1}^{l} v_{i_j,f}^{(l)} \right) \qquad (5)\end{aligned}$$

The straight-forward complexity for computing eq. (5) is $O(k_d n^d)$, but as done before for eq.(1), one can show that it can be computed in linear time.

1.2.5 FMS vs. SVMS

The table below compares the two models FMs and SVMs.

SVMs	**FMs**
Non-linear SVMs are learned in the dual.	Can be directly learned in the primal.
Prediction depends on parts of the training data.	Model equation is independent of the training data.
Model nested interactions between variables (while using a polynomial kernel) : The weights given to the interactions between variable i and j ($w_{i,j}$) is independent of the weight given to the interaction between variable i and k ($w_{i,k}$).	Model nested interactions between variables : The weights are factorized, ie $w_{i,j}$ and $w_{i,k}$ depend on each other as they overlap and share parameters due to the factorization.
Function best on dense data — that is, data with few to no missing values .	Estimate interactions in problems with a high degree of sparsity.
The time complexity depends on the number of free support vectors and the number of training samples.	Linear time complexity.

This comparison shows that the FMs were designed to address the weaknesses of SVMs.

1.2.6 Other Types of Factorization Models vs. Factorization Machines

A variety of factorization models are available:

- Standard models for m-ary relations on categorical variables such as MF, PARAFAC, etc.
- Specialized models for specific data and tasks such as SVD++, PITF, FPMC.

Some of them will be defined below:

- **Matrix and Tensor Factorization:** factorizes a relationship between two categorical variables (user-item) by defining binary indicator variables for each level of user and item.

- **SVD++:** this model takes into account implicit interactions (user and item bias) and the implicit feedback present in the set of rated items.
- **PITF for Tag Recommendation** models the two-way interactions between users, tags and items by factorizing each of the three relationships.
- **Factorized Personalized Markov Chains (FPMC):** models the matrix factorization of long-term user preference and markov chains of short-term sequential dynamics . It factorizes two matrices: the user-item matrix and the item-item transition matrix.

In the table below, we showed briefly the distinction between standard factorization models, Specialized factorizationmodels and FMs.

FMs	**Standard factorization models**	**Specialized factorization models**
Can work with any real-valued feature vector.	Require feature vectors divided in n groups (when there is n categorical variables) and in each group exactly one value has to be 1 and the rest 0	Designed for a single task.

Rendle [1] showed that FMs can emulate many of the most successful factorization models just by feature extraction. Therefore, it is sufficient to develop a single learning algorithm for FM to automatically obtain learning algorithms for the other factorization models.

[1] *See Steffen Rendle's paper for more details [5]*

1.2.7 Use-cases of FMs

Factorization machines models are often used for prediction tasks especially in the field of recommendation systems and click-through rate, some of them are listed here:

- Use of the Matrix Factorization (MF) in the Netflix Challenge to predict user ratings for movies based on previous ratings. [1]
- Exploiting Social Media for Stock Market Prediction. [2]
- Knowledge tracing, which consists on predicting the outcomes of students over questions as they are interacting with a learning platform. [3]
- ECML/PKDD discovery challenge for tag recommendation. [4]
- Factorized Personalized Markov Chains for Next Basket Recommendation (next purchase based on the past). [11]

1.2.8 Limits of FMs

- Because of the sparse setting, higher-order interactions are difficult to estimate, therefore, so far, most FM-related work focuses only on second-order (d=2).
- FM models present interactions in a linear way, which may be insufficient to capture the inherent non-linear and complex structure of real-world data.

1.3 Conclusion

In this report, we introduced Factorization Machines, a class of popular algorithms adopted for recommendation tasks. Their function as general predictors, their usage of the product of factorized parameters to model pairwise feature interactions,and their linear complexity make it highly expressive and powerful. **Factorization Machines is a must-have algorithm in any data scientist's pocket !**

References:

[1], [2], [3], [4], [5], [6], [7], [8], [9],[10],[11]