Accurate and Interpretable Factorization Machines

Liang Lan
Department of Computer Science,
Hong Kong Baptist University,
Hong Kong SAR, China
lanliang@comp.hkbu.edu.hk

Yu Geng
Department of Computer Science
and Technology,
East China Normal University, China
gydatoow@163.com

Abstract

Factorization Machines (FMs), a general predictor that can efficiently model high-order feature interactions, have been widely used for regression, classification and ranking problems. However, despite many successful applications of FMs, there are two main limitations of FMs: (1) FMs consider feature interactions among input features by using only polynomial expansion which fail to capture complex nonlinear patterns in data. (2) Existing FMs do not provide interpretable prediction to users. In this paper, we present a novel method named Subspace Encoding Factorization Machines (SEFM) to overcome these two limitations by using non-parametric subspace feature mapping. Due to the high sparsity of new feature representation, our proposed method achieves the same time complexity as the standard FMs but can capture more complex nonlinear patterns. Moreover, since the prediction score of our proposed model for a sample is a sum of contribution scores of the bins and grid cells that this sample lies in low-dimensional subspaces, it works similar like a scoring system which only involves data binning and score addition. Therefore, our proposed method naturally provides interpretable prediction. Our experimental results demonstrate that our proposed method efficiently provides accurate and interpretable prediction.

Introduction

Feature interactions play an important role in many machine learning algorithms for capturing nonlinear patterns in data. One popular example of using feature-interaction is Support Vector Machine (SVM) with polynomial kernel (Cortes and Vapnik 1995). However, implicit polynomial mapping via kernel trick induces huge computational cost since the number of support vectors in the SVM model increases linearly with the dataset size (Zhang et al. 2012). This is also known as the curse of kernelization (Wang, Crammer, and Vucetic 2012). Efficient approaches (Chang et al. 2010; Sonnenburg and Franc 2010) were proposed to explicitly pre-compute the low-degree polynomial kernel mapping and then apply linear SVM on mapped data. However, the number of features in polynomial kernel mapping scales as $O(d^q)$, where d is the number of input features and q is the degree of polynomial kernel (i.e. the degree of feature interaction). Therefore, these approaches only work on low degree feature interaction. To overcome this issue, FMs (Rendle 2010) were proposed to model feature interactions using factorized parameters. FMs can model high-degree feature interaction in linear time $O(d)$. In recent years, FMs have been widely used for many classification, regression and ranking problems.

Despite many successful applications of FMs, there are two main limitations of FMs: (1) FMs consider feature interactions among input features by using only polynomial expansion; (2) FMs do not provide interpretable prediction to user. With regards to the first limitation, FMs fail to capture complex nonlinear patterns in data. For example, many classification problems are not linear separately after polynomial feature mapping. Recently, Locally Linear Factorization Machine (LLFM) (Liu et al. 2017) was proposed to learn a complex nonlinear model by exploring local coding technique. They formulated a joint optimization to learn anchor points, local coordinates and FMs parameters together. However, due to the procedures of searching and updating local coding coordinates, LLFM requires high computational cost compared with standard FMs.

Apart from unable to capture complex nonlinear patterns, another limitation of FMs is the model interpretability. The feature interactions in FMs are modeled by polynomial expansion which are not easy to explain to users. In past few year, interpretability of machine learning models has attracted great research attention (Ribeiro, Singh, and Guestrin 2016; Chu et al. 2018; Chen et al. 2018). (Lou, Caruana, and Gehrke 2012) proposed Generalized Additive Models (GAM) to provide interpretable prediciton. The prediction of GAM is a sum of univariate models built on individual features. GAM can be explained because the user can visualize the contribution scores of individual input features (computed by univariate models) to final prediction by histogram. However, GAM does not consider feature interactions. GAM was further extended to Generalized Additive Models Plus Interactions (GA^{2}M) (Lou et al. 2013) by adding pairwise feature interactions among input features. The contribution scores of pairwise interactions to final prediction are visualized as heatmap in two-dimensional plane for interpretability. To avoid high computational cost of considering all pairwise feature interactions, GA^{2}M uses a heuristic method to select highly infor-

mative pairwise feature interactions. Fast Flux Discriminant (FFD) model (Chen, Chen, and Weinberger 2014) used a non-parametric subspace mapping method to model feature interactions. FFD first explicitly computes the mapped feature vectors by considering all possible one-dimensional and two-dimensional subspaces and then uses submodular optimization to select informative subspaces. Fully Corrective Binning (FCB) (Sokolovska, Chevaleyre, and Zucker 2018) was proposed to learn a scoring system from data. FCB simultaneously learns the interval threshold for data binning and also the associated contribute score for each bin. However, FCB only considers individual features and does not model feature interactions. It may result in suboptimal solutions in many problems since feature interactions are important for capturing nonlinear patterns in data.

In this paper, we propose a new method named Subspace Encoding Factorization Machines (SEFM). SEFM overcomes the aforementioned two limitations of standard FMs based on the following contributions. First, we achieve accurate and interpretable prediction by applying element-wise nonlinear feature mapping for both individual features and feature interactions in standard FMs. We first cut the low-dimensional subspaces formed by both individual features and feature interactions into bins and grid cells [1]. Each bin (or grid cell) is associated with a contribution score for prediction. We obtain the new feature mapping for both individual features and feature interactions using one-hot encoding. The non-zero entries in the new feature representation denote the contribute scores of different bins and grid cells that the samples lies in low-dimensional subspaces. The final prediction score of a sample is a sum of contribution scores of bins and grid cells this sample lies in low-dimensional subspaces. Therefore, our proposed model works like a scoring system that only involves data binning and score addition. It naturally gives interpretable predictions. Unlike the scoring system learnt by FCB (Sokolovska, Chevaleyre, and Zucker 2018) which only focus on individual features, our model can model high-order feature interactions.

However, explicitly computing one-hot encoded feature vectors for all feature interactions is impossible since it scales to $O(d^q)$. Our second contribution is to reformulate our proposed model as standard FMs on a new feature representation obtained by one-hot encoding on one-dimensional subspaces only. We provide a theoretical analysis to prove the equivalence. Therefore, our proposed model can be solved efficiently using existing FMs tools. Moreover, due to the high sparsity of the new feature representation, our proposed model achieves linear time complexity $O(d)$, which is the same as the standard FMs.

Finally, we performed extensive experiments to evaluate our proposed algorithm on both synthetic and real-life benchmark datasets. Our experimental results clearly demonstrate the effectiveness, efficiency and interpretability of our proposed method. On all datasets, our proposed model always gets higher accuracy than standard FMs. The accuracies obtained by our model is close to and sometimes higher than SVM with rbf kernel, which clearly indicates the proposed model can capture complex nonlinear patterns in data. We also demonstrate the efficiency and interpretability of our proposed model in the experiments section.

[1] In this paper, the meanings of "bin" and "grid cell" are the same. We refer to "bin" in one-dimensional subspaces and "grid cell" in high-dimensional subspaces.

Methodology

Preliminary: Factorization Machines

FMs (Rendle 2010) are highly related to our proposed method. In this section, we introduce our notations and review standard FMs. Assume we are given a training data set $D = \{\mathbf{x}_i, y_i\}_{i=1}^n$, where $\mathbf{x}_i$ is a d dimensional input feature vector, $y_i \in \{-1, 1\}$ is the corresponding class label. In this paper, we use x_{ij} to denote the j-th feature value of the i-th sample and use $\mathbf{x}^j$ to denote the values of feature j across all samples. FMs with degree-2 are defined as:

$$f(\mathbf{x}_i) = \sum_{j=1}^{d} w_j x_{ij} + \sum_{j=1}^{d} \sum_{k=j+1}^{d} \tilde{w}_{jk} x_{ij} x_{ik}, \tag{1}$$

where w_j denotes the model coefficient for the j-th feature and $\tilde{w}_{jk}$ denotes the model coefficient for the pairwise feature interaction between feature j and feature k. In FM, the $\tilde{w}_{jk}$ is factorized as $\mathbf{v}_j^T \mathbf{v}_k$, where $\mathbf{v}_j$ is an m-dimensional vector. According to Lemma 3.1 in (Rendle 2010), the second term in (1) can be computed in linear time $O(md)$ as shown in following

$$\begin{aligned} \sum_{j=1}^{d} \sum_{k=j+1}^{d} \tilde{w}_{jk} x_{ij} x_{ik} &= \sum_{j=1}^{d} \sum_{k=j+1}^{d} \langle \mathbf{v}_j^T \mathbf{v}_k \rangle x_{ij} x_{ik} \\ &= \frac{1}{2} \sum_{f=1}^{m} ((\sum_{j=1}^{d} v_{j,f} x_{ij})^2 - \sum_{j=1}^{d} v_{j,f}^2 x_{ij}^2). \end{aligned} \tag{2}$$

Therefore, the model parameters of FMs (i.e., $\mathbf{w} \in \mathbb{R}^d$ and $\mathbf{V} \in \mathbb{R}^{d \times m}$) can be efficiently learnt by Stochastic Gradient Descent (SGD). The gradient of FM based on one sample $\mathbf{x}_i$ is

$$\begin{cases} \frac{\partial f(\mathbf{x}_i)}{\partial w_j} = x_{ij} \\ \frac{\partial f(\mathbf{x}_i)}{\partial v_{jf}} = x_{ij} \sum_{f=1}^{d} v_{jf} x_{ij} - v_{jf} x_{ij}^2. \end{cases} \tag{3}$$

Compared with SVM with polynomial-2 kernel (Cortes and Vapnik 1995), the main advantage of FMs is to reduce the time complexity of modeling pairwise feature interactions from $O(d^2)$ to $O(md)$ by using low rank factorization of $\tilde{w}_{jk} = \mathbf{v}_j^T \mathbf{v}_k$. Many off-the-shelf tools (Rendle 2010; Bayer 2016) have been developed to learn FMs parameters from training data.

Subspace Encoding Factorization Machines

Despite many successful applications of FMs, there are two main limitations of FMs: (1) FMs consider feature interactions among input features by using only polynomial expansion which fail to capture complex nonlinear patterns in data; (2) Existing FMs do not provide interpretable prediction to users. To overcome these two limitations, we present our proposed algorithm based on subspace encoding in this

section. Our idea is to apply element-wise feature mapping for both individual features and feature interactions in standard FMs as shown in (1). In other words, we map each individual feature value x_{ij} to a high dimensional vector $\Phi(x_{ij})$ and map each feature interaction value $x_{ij}x_{ik}$ to a high dimensional vector $\Phi(x_{ij}, x_{ik})$. Without loss of generality, we focus our discussion on degree-2 FMs in the rest of this paper. However, our proposed method can be easily generalized to high degree FMs. By element-wise feature mapping, the standard FMs (1) can be rewritten as

$$f(\mathbf{x}_i) = \sum_{j=1}^{d} \mathbf{w}_j \Phi(x_{ij}) + \sum_{j=1}^{d} \sum_{k=j+1}^{d} \tilde{\mathbf{w}}_{jk} \Phi(x_{ij}, x_{ik}). \quad (4)$$

Now, the key question is how to construct efficient and interpretable $\Phi(x_{ij})$ and $\Phi(x_{ij}, x_{ik})$. Motivated by recent work on learning scoring system from data (Sokolovska, Chevaleyre, and Zucker 2018) and FFD (Chen, Chen, and Weinberger 2014). We propose to construct the feature mapping by first cutting input space into different bins and grid cells in low-dimensional subspaces. Then, we use one-hot encoding to construct both $\Phi(x_{ij})$ and $\Phi(x_{ij}, x_{ik})$. The non-zero entry in $\Phi(x_{ij})$ denotes the contribution score of the bin that sample $\mathbf{x}_i$ lies in the one-dimensional subspace formed by feature j. Similarly, the non-zero entry in $\Phi(x_{ij}, x_{ik})$ denotes the contribution score of the grid cell that sample $\mathbf{x}_i$ lies in the two-dimensional subspace formed by feature j and feature k. We illustrate the idea of subspace feature encoding in Figure 1.

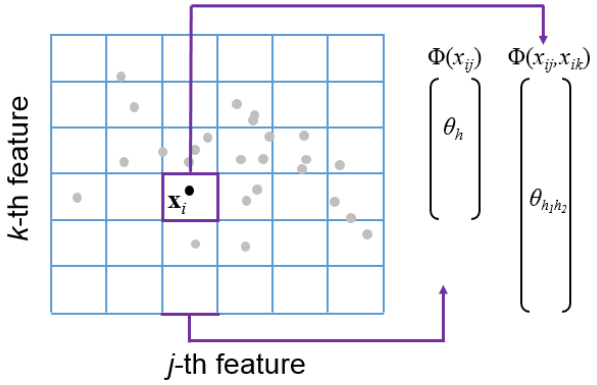


Figure 1: Subspace Feature Encoding for FM

As shown in Figure 1, we first divide each dimension into multiple bins with equal length. Suppose each dimension is divided into b bins for simplicity [2]. For a numerical feature j, let us use $\min\{\mathbf{x}^j\}$ to denote the minimal value and use $\max\{\mathbf{x}^j\}$ to denote the maximal value of feature j. Then the interval boundary of the h-th ($1 \leq h \leq b$) bin for feature j is

$$B_h^j = \begin{cases} [l_h^j, u_h^j) & \text{if } h < b \\ [l_h^j, u_h^j] & \text{if } h = b \end{cases}, \quad (5)$$

[2]For a categorical feature, the b is equal to the number of unique categorical values in this feature

where

$$l_h^j = \min\{\mathbf{x}^j\} + \frac{\max\{\mathbf{x}^j\} - \min\{\mathbf{x}^j\}}{b}(h-1) \quad (6)$$

and

$$u_h^j = \min\{\mathbf{x}^j\} + \frac{\max\{\mathbf{x}^j\} - \min\{\mathbf{x}^j\}}{b}h. \quad (7)$$

Definition 1. FM feature mapping. We define our one-hot feature mapping for individual feature and feature-interaction as

$$\begin{aligned} \Phi(x_{ij}) &= [0, \ldots, \theta_h, \ldots, 0] \\ \Phi(x_{ij}, x_{ik}) &= [0, \ldots, \tilde{\theta}_{h_1h_2}, \ldots, 0]. \end{aligned} \quad (8)$$

$\Phi(x_{ij})$ is a one-hot vector with length b, where the non-zero entry θ_h indicates $x_{ij} \in B_h^j$. Similarly, $\Phi(x_{ij}, x_{ik})$ is a one-hot vector with length b^2, where the non-zero entry $\tilde{\theta}_{h_1h_2}$ indicates $x_{ij} \in B_{h_1}^j$ and $x_{ik} \in B_{h_2}^k$. In other words, one-hot encoded feature vector of a sample tells us both the indices of bins (e.g. h) and grid cells of this sample lies in low-dimensional subspaces and the corresponding contribution scores of these bins (e.g. θ_h) and grid cells. By using (8), FMs are able to capture complex nonlinear patterns in data. In the meanwhile, the element-wise feature mapping $\Phi(x_{ij})$ and $\Phi(x_{ij}, x_{ik})$ can be easily traced back to original input features and non-zero values corresponding to contribution scores can be easily explained to user. The interpretability of our proposed model will be further discussed in a later section.

However, explicitly computing $\Phi(x_{ij}, x_{ik})$ for all feature interactions is impractical because it scales to $O(d^2)$ and d usually is large for real-life applications. To overcome this issue, in the following, we show our proposed model in (4) can be reformulated as standard FMs on a new feature representation obtained by using one-hot encoding on one-dimensional subspaces only. We also provide a theoretical analysis to prove the equivalence.

Definition 2. One-hot encoding on one-dimensional subspaces. For any input $\mathbf{x}_i$, we define the following one-hot encoding on one-dimensional subspaces formed by individual input features.

$$z_{ijh} = \begin{cases} 1, & \text{if } x_{ij} \in B_h^j \\ 0, & \text{otherwise.} \end{cases} \quad (9)$$

Therefore, $\mathbf{Z}$ can be viewed as a matrix with size $n \times (b \times d)$. Each column in $\mathbf{X}$ is replaced by b columns in $\mathbf{Z}$. The new feature representation $\mathbf{Z}$ only contains zeroes and ones. The key advantage of this binary one-hot encoding is that the non-zero entry z_{ijh} tells us the bin index of the i-th sample lies in the one-dimensional subspace formed by feature j. In other words, $z_{ijh} = 1$ tells us that the i-th sample is located in the h-th bin of the one-dimensional subspace formed feature j. Moreover, the pairwise feature interaction is also a binary one-hot vector where non-zero value $z_{ijh_1}z_{ikh_2} = 1$ tells us the grid cell indices of the i-th samples lies in the two-dimensional subspace formed by feature j and feature k (i.e., the i-th sample is located in the intersection of the h_1-th bin of feature j and the h_2-th bin of feature k in the

two-dimensional subspace formed by feature j and feature k.).

Proposition 1. *Our proposed model defined in (4) together with feature mapping defined in (8) is equivalent to standard FMs on* $\mathbf{Z}$ *as defined in (9)*

Proof. By definition of $\Phi(x_{ij}), \Phi(x_{ij}, x_{ik})$ and $\mathbf{Z}$, (8) can be rewritten as

$$\begin{aligned}\Phi(x_{ij}) &= [z_{ij1}\theta_1, z_{ij2}\theta_2, \ldots, z_{ijh}\theta_h, \ldots, z_{ijb}\theta_b] \\ \Phi(x_{ij}, x_{ik}) &= [z_{ij1}z_{ik1}\tilde{\theta}_{11}, \ldots, \\ &\qquad z_{ijh_1}z_{ikh_2}\tilde{\theta}_{h_1h_2}, \ldots, z_{ijb}z_{ikb}\tilde{\theta}_{bb}].\end{aligned} \tag{10}$$

By plugging (10) into (4), we obtain

$$\begin{aligned}f(\mathbf{x}_i) = &\sum_{j=1}^{d}\sum_{h=1}^{b} w_h^j\theta_h z_{ijh} + \\ &\sum_{j=1}^{d}\sum_{k=j+1}^{d}\sum_{h_1=1}^{b}\sum_{h_2=1}^{b} \tilde{w}_{h_1h_2}^{jk}\tilde{\theta}_{h_1h_2}^{jk} z_{ijh_1}z_{ikh_2}.\end{aligned} \tag{11}$$

By treating $w_h^j\theta_h$ together as a single variable β_h^j and treating $\tilde{w}_{h_1h_2}^{jk}\tilde{\theta}_{h_1h_2}^{jk}$ together as a single variable $\tilde{\beta}_{h_1h_2}^{jk}$, (11) can be rewritten as

$$\begin{aligned}f(\mathbf{x}_i) = &\sum_{j=1}^{d}\sum_{h=1}^{b} \beta_h^j z_{ijh} + \\ &\sum_{j=1}^{d}\sum_{k=j+1}^{d}\sum_{h_1=1}^{b}\sum_{h_2=1}^{b} \tilde{\beta}_{h_1h_2}^{jk} z_{ijh_1}z_{ikh_2}.\end{aligned} \tag{12}$$

In the other hand, FM model on $\mathbf{Z}$ is

$$f(\mathbf{z}_i) = \sum_{l_1=1}^{d*b} w_j z_{il_1} + \sum_{l_1=1}^{d*b}\sum_{l_2=l_1+1}^{d*b} \tilde{w}_{l_1l_2} z_{il_1}z_{il_2}. \tag{13}$$

By following index notation in (9), z_{il_1} in (13) equals to z_{ijh_1} in (12) where $j = \lfloor l_1/b \rfloor$ and $h_1 = l_1 \% b$. Similarly, z_{il_2} in (13) equals z_{ikh_2} where $k = \lfloor l_2/b \rfloor$ and $h_2 = l_2 \% b$. Therefore, it is straight forward to verify that the first term in (13) is in the same form as the first term in (12), i.e., $\sum_{l_1=1}^{d*b} w_j z_{il_1} = \sum_{j=1}^{d}\sum_{h=1}^{b} w_h^j z_{ijh}$. And the second term in (13) can be rewritten as

$$\begin{aligned}&\sum_{l_1=1}^{d*b}\sum_{l_2=j+1}^{d*b} \tilde{w}_{l_1l_2} z_{il_1}z_{il_2} = \\ &\sum_{j=1}^{d}\sum_{k=j+1}^{d}\sum_{h_1=1}^{b}\sum_{h_2=1}^{b} \tilde{w}_{h_1h_2}^{jk} z_{ijh_1}z_{ikh_2} \\ &+ \sum_{j=1}^{d}\sum_{h1=1}^{b}\sum_{h_2=h_1+1}^{b} \tilde{w}_{h_1h_2}^{jj} z_{ijh_1}z_{ijh_2}.\end{aligned} \tag{14}$$

Note that $z_{ijh_1}z_{ijh_2} = 0$ as long as $h_1 \neq h_2$. Therefore, $\sum_{j=1}^{d}\sum_{h_1=1}^{b}\sum_{h_2=h_1+1}^{b} \tilde{w}_{h_1h_2}^{jj} z_{ijh_1}z_{ijh_2}$ always equals to 0 and can be removed from (14). Finally, (13) can be written as

$$\begin{aligned}f(\mathbf{z}_i) = &\sum_{j=1}^{d}\sum_{h=1}^{b} w_h^j z_{ijh} + \\ &\sum_{j=1}^{d}\sum_{k=j+1}^{d}\sum_{h_1=1}^{b}\sum_{h_2=1}^{b} \tilde{w}_{h_1h_2}^{jk} z_{ijh_1}z_{ikh_2}.\end{aligned} \tag{15}$$

Comparing (15) with (12), we can see that these two problems are in the same form and will have the same solution. In conclusion, our proposed model defined in (4) together with feature mapping defined in (8) is equivalent to standard FMs on $\mathbf{Z}$ as defined in (9). □

Proposition (1) clearly shows that our proposed model in (4) can be solved efficiently by applying standard FMs on $\mathbf{Z}$ as defined in (9).

Interpretability of Our Proposed Algorithm

To show the interpretability of our proposed algorithm, let us consider our model in the form of (12). The first term $\sum_{j=1}^{d}\sum_{h=1}^{b} \beta_h^j z_{ijh}$ computes a sum of the contribution scores in one-dimensional subspaces formed by individual features in original input space. The non-zero entry z_{ijh} in one-hot vector $\mathbf{z}_{ij}$ with length b tells us the bin index (i.e. the h-th bin) of the i-th sample lies in the one-dimensional subspace formed by feature j. And β_h is the contribution score of this particular bin. Similarly, $\sum_{j=1}^{d}\sum_{k=j+1}^{d}\sum_{h_1=1}^{b}\sum_{h_2=1}^{b} \tilde{\beta}_{h_1h_2}^{jk} z_{ijh_1}z_{ikh_2}$ computes a sum of the contribution scores in two-dimensional subspaces. The non-zero entry $z_{ijh_1}z_{ikh_2}$ tells us grid cell indices of the i-th sample in two-dimensional subspace formed by feature j and feature k. And $\tilde{\beta}_{h_1h_2}^{jk}$ is the contribution score of this particular grid cell. Therefore, the final prediction score of a sample is a sum of contribution scores of the bins and grid cells that this sample lies in low-dimensional subspaces. This prediction score can be easily explained to users since it works like scoring system (Still et al. 2014) that only involves data binning and score addition. Furthermore, similar to GA^{2}M, we can visualize contribution scores of each individual feature by histogram and visualize contribution scores of each feature interaction by heatmap for model interpretation.

Algorithm Implementation and Analysis

According to Proposition (1), our proposed model in (4) is equivalent to apply standard FM on $\mathbf{Z}$ as defined in (9). Therefore, our proposed can be easily implemented based on existing FMs tools. We name our algorithm as Subspace Encoding Factorization Machines (SEFM) and summarize it in Algorithm 1. According the definition of $\mathbf{Z}$ in (9), $\mathbf{Z}$ is a very sparse matrix. For encoding feature j in $\mathbf{x}_i$, we only need to figure out the bin index of sample $\mathbf{x}_i$ in one-dimensional subspace formed by feature j. This index can be computed by

$$h = \min\{\left\lfloor \frac{\{x_{ij} - \min\{\mathbf{x}^j\}\}}{\max\{\mathbf{x}^j\} - \min\{\mathbf{x}^j\}} b \right\rfloor + 1, b\}. \tag{16}$$

This computation only needs $O(1)$ time. Therefore, the step 1 in Algorithm 1 only requires $O(nd)$ time and the required time is independent with the number of bins b. Due to the nature of one-hot encoding, the number of non-zero values in new representation of i-th sample $\mathbf{z}_i$ is d. Therefore, the step 2 of training an FM model using SGD only requires $O(nmd)$ time, which is the same as standard FMs when the input data matrix is dense. If the input data matrix is sparse, our proposed algorithm will be a few times slower than standard FMs depending on the sparsity level of the input data. The prediction time of our model is $O(md)$, which is very efficient compared with other nonlinear classifiers.

Algorithm 1 Subspace Encoding Factorization Machines

Training

Input: training data set $D = \{\mathbf{x}_i, y_i\}_{i=1}^n$, low-rank parameter m, number of bins b, regularization parameter C;
Output: FM model on mapped feature space;

1: Generate new feature representation $\mathbf{Z}$ as defined in (9)
2: Build an FM model $f(\mathbf{z})$ using $\{\mathbf{z}_i, y_i\}_{i=1}^n$

Related Works

In this section, we discuss the relationships of our proposed algorithms with other work.

Fully Corrective Binning (FCB). FCB was proposed by (Sokolovska, Chevaleyre, and Zucker 2018) for learning interpretable scoring system from data. Their basic idea is to automatically bin input data and obtain the corresponding contribution score for each bin. The prediction score of a sample is computed as a sum of contribution scores of the bins that this sample lies in. The proposed learning algorithm for FCB works in an iterative manner. In each iteration, it greedily finds the optimal binning and the contributions scores of different bins. Our proposed algorithm is also motivated by scoring system. However, compared with FCB, our algorithm has several important differences. First, FCB only considers binning individual features and ignores feature interactions while our algorithm considers all pair-wise feature interactions. Second, FCB uses a greedy method to find the optimal binning and the corresponding scores of different bins. In comparison, we use equal-length bins and learn the contribution scores of different bins and grid cells by using FMs. The greedy schema used in FCB may result in suboptimal solutions. Also FCB requires more computational time than our method.

Fast Flux Discriminant (FFD). FFD model uses a linear logistic regression model on top of a feature representation obtained by using histogram estimation in low-dimensional subspaces. It uses submodular optimization to select informative subspaces and use l_1 regularization to learn a sparse model for interpretability. Our proposed method is related to FFD on the subspace feature encoding. However, FFD used histogram estimation to compute the new feature values (similar to contribution scores in our method) for each bin/grid cell. The procedures of feature value learning and classification model training in FFD are decoupled. In comparison, our proposed method couples new feature values learning and classification model training together. By coupling them together, our method could achieve more accurate prediction. Another difference is efficiency. FFD needs to explicitly compute new feature representation based on all possible one-dimensional and two-dimensional subspaces. It scales to $O(d^2)$ whereas our proposed method achieve linear time complexity $O(d)$ by applying FMs on one-hot encoded feature representation using one-dimensional subspaces only.

Generalized Additive Models Plus Interactions (GA^{2}M). GA^{2}M model was proposed to extend GAM by considering feature interactions. To avoid huge computational cost caused by a large number (i.e. $O(d^2)$) of pairwise feature interactions, they proposed a heuristic-based method to select a limited number of informative pairwise feature interactions. The prediction score GA^{2}M is a sum of contribution scores based on both on individual input features and selected feature interactions. Our method also uses similar idea for computing prediction score. However, the contribution scores in GA^{2}M is estimated by using tree-based or spline-based models whereas the contribution scores in our model are learnt by FMs model. Again, GA^{2}M used greedy based method to select a few pairwise feature interactions whereas our proposed model uses FM model to efficiently consider all pairwise feature interactions.

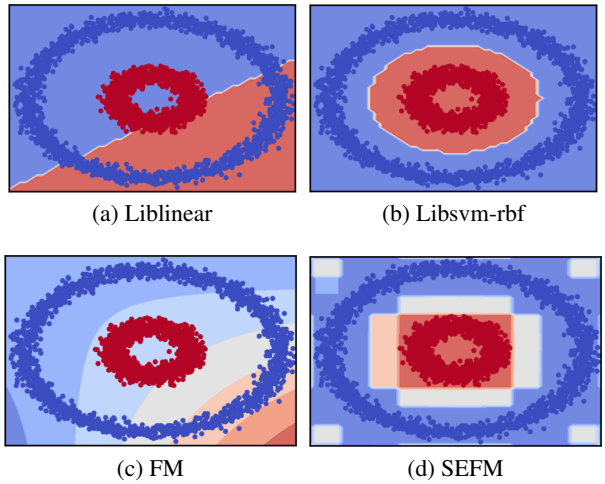
Experiments

In this section, we compare our proposed method with competing approaches on two synthetic datasets and five benchmark datasets.

In our experiments, we evaluate the performance of the following four algorithms:

- Liblinear: an efficient solver for linear support vector machine (Fan et al. 2008);
- Libsvm-rbf: support vector machine with rbf kernel (Chang and Lin 2011);
- FM: factorization machines (Rendle 2010);
- LLFM: Locally Linear Factorization Machines (Liu et al. 2017);
- SEFM: our proposed method.

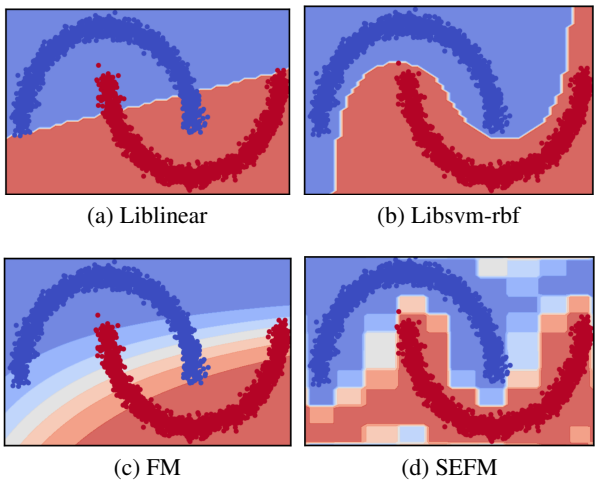
Synthetic Datasets. We first use two synthetic nonlinear datasets circles and moons to illustrate how our proposed method cuts the input feature space and builds the interpretable nonlinear classifier. Circles is a binary classification dataset in two-dimensional space as shown in Figure 2. The blue points in the large outer circle belong to one class and the red points in the inner circle belong to the other class. Moons is also a two-dimensional binary classification dataset. It represents two interleaving half circles as shown in Figure 3. We compare the performance of listed five algorithms on both circles and moons datasets and the results are reported in Table (1). We also show the decision boundaries of Liblinear, Libsvm-rbf, FM and SEFM in Figure (2) for circles dataset and Figure (3) for moons dataset.



(a) Liblinear (b) Libsvm-rbf

(c) FM (d) SEFM

Figure 2: Comparison of the decision boundaries of four different classifiers on circles data



(a) Liblinear (b) Libsvm-rbf

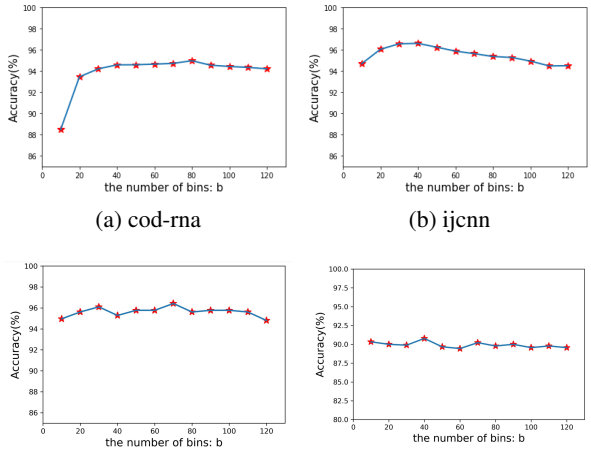
(c) FM (d) SEFM

Figure 3: Comparison of the decision boundaries of four different classifiers on moons data

As shown in these two figures, the linear classifier (i.e. Liblinear) can not handle these two nonlinear classification problems. FM can produce nonlinear decision boundaries. However, due to that the standard FMs consider feature interactions by using only degree 2 polynomial expansion, the produced decision boundaries (i.e. sub-figure (c)) are far from the optimal one. Both LibSVM-rbf (sub-figure (a)) and SEFM (sub-figure (d)) can perfectly separate these two datasets. Since our proposed model uses low-dimensional subspace encoding, it produces piecewise axis perpendicular decision boundary. Each bin (or grid cell) is associated with a contribution score that can be easily explained to users.

Evaluations on Benchmark Datasets. In addition to two synthetic datasets, we also evaluate the performance of the five algorithms on other five benchmark datasets. These five datasets are publicly available at the Libsvm website [3]. We report our experimental results in Table (1). The summary of each dataset (i.e., number of samples, number of features and number of classes) is given in the first column of the table. For each dataset, we randomly select 70% as training data and use the remaining 30% as test data. The process is repeated 10 times and we report the average accuracy on test data. For all five algorithms, the regularization parameter is chosen from $\{10^{-3}, 10^{-2}, \ldots, 10^{2}, 10^{3}\}$. For Libsvm with rfb kernel, the kernel width is chosen from $\{2^{-5}, 2^{-4}, \ldots, 2^{4}, 2^{5}\}$. The low-rank parameter m for FM, LLFM and SEFM is chosen from {2, 4, 8, 16, 32, 64}. The parameter b (i.e., the number of bins) of SEFM is chosen from $\{10, 20, 30, \ldots, 120\}$. The optimal parameter combination is selected by 5-fold cross-validation on training data.

The accuracy and running time are reported in Table (1). With respect to accuracy, our proposed method obtains higher accuracy than both Liblinear and FM on all seven datasets. Compared with Libsvm with rbf kernel, our proposed method achieves comparable classification accuracies on circles, moons, breast-cancer and cod-rna datasets. In the other three datasets, our proposed method SEFE gets better accuracies than Libsvm with rbf kernel. Compared with LLFM, our method has similar accuracies on the two synthetic datasets. LLFM only gets better accuracy than our method on breast-cancer dataset. For the other four benchmark datasets, our proposed method gets better accuracies than LLFM. With respect to running time, our proposed method is fast. As expected, the running time of our proposed method is comparable to standard FMs in all datasets except webspam. For webspam dataset, the original data is sparse text data. Therefore, our proposed algorithm is slower than FM in this dataset. LLFM is slower than FM



(a) cod-rna (b) ijcnn

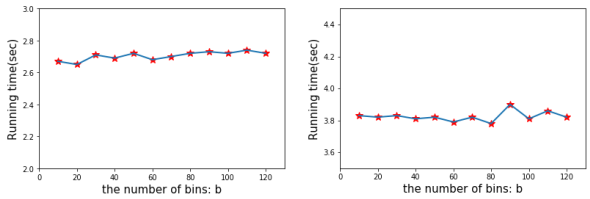
(c) breast-cancer (d) splice

Figure 4: Classification accuracy vs. the number of bins (b)

[3] https://www.csie.ntu.edu.tw/ cjlin/libsvmtools/datasets/

Table 1: The accuracy and running time (in seconds) (The best results are in bold)

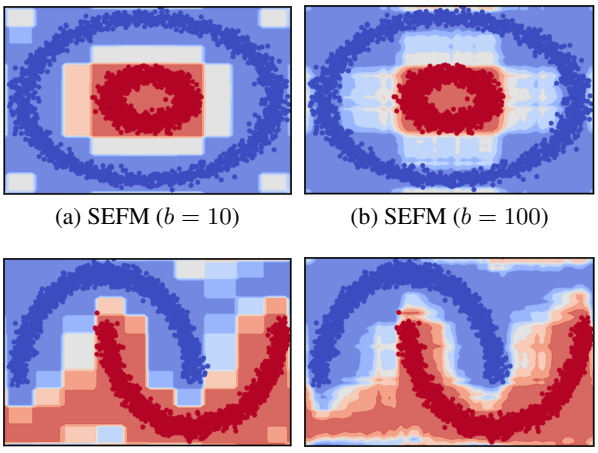
Dataset n/d/#class	Performance	Liblinear	SVM-rbf	FM	LLFM	SEFM
circles 5,000/2/2	accuracy(%) time	48.99±1.29 0.02	**99.99±0.00** 0.17	49.70±1.85 0.24	**99.99±0.00** 8.2	99.95±0.03 0.22
moons 5,000/2/2	accuracy(%) time	88.51±0.11 0.03	**99.99±0.00** 0.19	87.53±0.09 0.24	**99.99±0.00** 8.1	**99.99±0.00** 0.23
breast-cancer 683/9/2	accuracy(%) time	95.12±0.98 0.01	95.93±0.87 0.01	94.63±0.52 0.14	**98.49±0.50** 2.9	96.59±0.97 0.14
splice 1,000/60/2	accuracy(%) time	80.33±1.63 0.02	87.22±1.39 0.13	82.17±0.22 0.39	88.03±1.72 8.4	**91.44±0.17** 0.36
ijcnn 49,990/22/2	accuracy(%) time	91.95±0.07 0.85	94.65±0.09 16.33	93.14±0.29 3.63	95.89±0.45 106.1	**96.42±0.32** 3.42
cod-rna 59,535/8/2	accuracy(%) time	92.65±0.04 1.77	95.03±0.03 20.12	93.36±0.61 3.0	85.41±2.18 100.7	**95.12±0.14** 2.70
webspam 350,000/254/2	accuracy(%) time	92.74±0.02 17	92.17±0.15 16531	93.83±0.17 116	92.92±0.34 3702	**96.02±0.10** 275.4



(a) cod-rna (b) ijcnn

(c) breast-cancer (d) splice

Figure 5: Runing time vs. the number of bins (b)



(a) SEFM ($b = 10$) (b) SEFM ($b = 100$)

(c) SEFM ($b = 10$) (d) SEFM ($b = 100$)

Figure 6: The decision boundaries of SEFM with different b

and SEFM due to the cost of searching and updating local coding coordinates.

Impact of the parameters b. In this section, we evaluate the impact of parameter b in our proposed algorithm SEFM. Figure (4) shows the accuracy of our proposed method SEFM on dataset cod-rna, ijcnn, breast-cancer and splice with respect to different b. As shown in Figure (4), the accuracy increases as parameter b increases when b is not large enough. Then, it will become stable very quickly. We also report the running time of our proposed method in Figure (5) with respect to different b, as expected, the running time of our proposed method does not increase as we increase parameter b. In Figure (6), we show how decision boundary changes on circles and moons data as we increase parameter b. We can see that the decision boundaries will become smoother when we increase parameter b.

Conclusion

In this paper, we propose a novel model to overcome the existing limitations of standard FMs by using subspace encoding. Our proposed method achieves the same time complexity as the standard FMs but can capture more complex nonlinear patterns in data. Moreover, the final prediction score of our proposed algorithm for a sample is a sum of the contribution scores of the bins and grid cells that this sample lies in. It works like a scoring system and naturally provides interpretable prediction. We evaluate the performance of our proposed model on both synthetic and real-life benchmark datasets. The experimental results clearly show our proposed method gets better accuracies than FMs. The accuracies obtained by our model is close and sometimes higher than SVM model with rbf kernel, which indicates our model can capture complex nonlinear patterns. We also demonstrate the efficiency and interpretability of our model.

Acknowledgments

This work was supported by the start-up fund from Department of Computer Science, Hong Kong Baptist University.

References

Bayer, I. 2016. fastfm: A library for factorization machines. *Journal of Machine Learning Research* 17(184):1–5.

Chang, C.-C., and Lin, C.-J. 2011. Libsvm: a library for support vector machines. *ACM transactions on intelligent systems and technology (TIST)* 2(3):27.

Chang, Y.-W.; Hsieh, C.-J.; Chang, K.-W.; Ringgaard, M.; and Lin, C.-J. 2010. Training and testing low-degree polynomial data mappings via linear svm. *Journal of Machine Learning Research* 11(Apr):1471–1490.

Chen, J.; Song, L.; Wainwright, M. J.; and Jordan, M. I. 2018. Learning to explain: An information-theoretic perspective on model interpretation. In *Proceedings of the 35th International Conference on Machine Learning*, 882–891.

Chen, W.; Chen, Y.; and Weinberger, K. Q. 2014. Fast flux discriminant for large-scale sparse nonlinear classification. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, 621–630. ACM.

Chu, L.; Hu, X.; Hu, J.; Wang, L.; and Pei, J. 2018. Exact and consistent interpretation for piecewise linear neural networks: A closed form solution. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 1244–1253.

Cortes, C., and Vapnik, V. 1995. Support-vector networks. *Machine learning* 20(3):273–297.

Fan, R.-E.; Chang, K.-W.; Hsieh, C.-J.; Wang, X.-R.; and Lin, C.-J. 2008. Liblinear: A library for large linear classification. *Journal of machine learning research* 9(Aug):1871–1874.

Liu, C.; Zhang, T.; Zhao, P.; Zhou, J.; and Sun, J. 2017. Locally linear factorization machines. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, 2294–2300. AAAI Press.

Lou, Y.; Caruana, R.; Gehrke, J.; and Hooker, G. 2013. Accurate intelligible models with pairwise interactions. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, 623–631. ACM.

Lou, Y.; Caruana, R.; and Gehrke, J. 2012. Intelligible models for classification and regression. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, 150–158. ACM.

Rendle, S. 2010. Factorization machines. In *IEEE 10th International Conference on Data Mining (ICDM)*, 995–1000.

Ribeiro, M. T.; Singh, S.; and Guestrin, C. 2016. Why should i trust you?: Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, 1135–1144. ACM.

Sokolovska, N.; Chevaleyre, Y.; and Zucker, J.-D. 2018. A provable algorithm for learning interpretable scoring systems. In *International Conference on Artificial Intelligence and Statistics*, 566–574.

Sonnenburg, S., and Franc, V. 2010. Coffin: a computational framework for linear svms. In *Proceedings of the 27th International Conference on International Conference on Machine Learning*, 999–1006.

Still, C. D.; Wood, G. C.; Benotti, P.; Petrick, A. T.; Gabrielsen, J.; Strodel, W. E.; Ibele, A.; Seiler, J.; Irving, B. A.; Celaya, M. P.; et al. 2014. A probability score for preoperative prediction of type 2 diabetes remission following rygb surgery. *The lancet. Diabetes & endocrinology* 2(1):38.

Wang, Z.; Crammer, K.; and Vucetic, S. 2012. Breaking the curse of kernelization: Budgeted stochastic gradient descent for large-scale svm training. *Journal of Machine Learning Research* 13(Oct):3103–3131.

Zhang, K.; Lan, L.; Wang, Z.; and Moerchen, F. 2012. Scaling up kernel svm on limited resources: A low-rank linearization approach. In *Artificial Intelligence and Statistics*, 1425–1434.