Check for updates

Sparse Ordinal Regression via Factorization Machines

Weijian Ni, Tong Liu(✉), and Qingtian Zeng

College of Computer Science and Engineering,
Shandong University of Science and Technology, Qingdao, China
niweijian@gmail.com, liu_tongtong@foxmail.com, qtzeng@163.com

Abstract. Most existing ordinal regression methods are adapted from traditional supervised learning algorithms (e.g., support vector machines and neural networks) which have shown to work well mostly on dense data. However, the use of existing ordinal regression methods on sparse data has received less scrutiny. This paper proposes to address the sparsity issue arose in many real-world ordinal regression applications by leveraging the feature interaction modeling techniques. Following the popular threshold methodology in ordinal regression studies, we extend Factorization Machines, an effective solution to modeling pairwise feature interactions in sparse feature space, to ordinal regression. The proposed model, namely Factorization Machines for Ordinal Regression (FMOR), combines the ability of threshold methodology in predicting targets of ordinal scale with the advantages of factorization models in handling high-dimensional sparse data. Through extensive experimental studies, we show that the proposed FMOR is both effective and efficient against state-of-the-art baselines.

Keywords: Ordinal regression · Factorization machines · Sparse data

1 Introduction

Ordinal regression is an important type of supervised learning paradigm, which aims to learn predictive models for ordinal targets. Ordinal regression problems are very common in massive domains from social sciences [9] to financial technology [1] and clinical research [23]. In recent years, ordinal regression has experienced significant developments, with many prevalent methods adapted from traditional machine learning algorithms such as support vector machines [3,24], neural networks [2,4,6,13], boosting [11] and discriminant learning [21,22]. These methods have shown to be effective in many scenarios, but unavoidably, retain substantial weaknesses of the original methods. One significant challenge comes from the fact that the feature space can be of very high dimension but sparse in many real-world ordinal regression applications, e.g., collaborative filtering, click-through rate prediction, and computer-aided pathology diagnosis. It is known that the sparse representation problem greatly hinders the performance of traditional machine learning methods, as well as their extensions for ordinal regression [7].

© Springer Nature Switzerland AG 2019
A. C. Nayak and A. Sharma (Eds.): PRICAI 2019, LNAI 11671, pp. 162–174, 2019.
https://doi.org/10.1007/978-3-030-29911-8_13

One successful solution to the sparse representation problem is to model the inherent interactions among features because co-occurrence of features often helps reveal high-level domain knowledge about the task under consideration. One effective approach to model feature interaction is Factorization Machines (FM) [19], which embeds high-dimensional sparse features into a rank-low latent space and learns pairwise feature interactions via the inner product of features' embedding vectors. Although originally proposed in the context of recommender systems, FM has yielded great promise in a wide range of prediction tasks, especially those with very high and sparse feature space [16–18,25]. However, the target variables of traditional FM models can only be either discrete or continuous. Thus, the tradition FM does not yet cater for the ordinal relationship among learning targets. To our best knowledge, there is little work adapting FM for ordinal regression.

In this paper, we propose a novel **F**actorization **M**achines for **O**rdinal **R**egression (FMOR), in which the sparsity issue in ordinal regression is tackled through factorized feature interactions. Motivated by the popular threshold methodology of ordinal regression studies, the proposed FMOR extends the traditional FM by introducing a set of threshold parameters that map real-valued outputs of FM to ordinal labels. We implement the learning algorithm of FMOR based on stochastic gradient descent, and further claim that the ordinal threshold constraint required by threshold-based ordinal regression methods can be automatically satisfied by the derived model. Finally, we perform comprehensive experiments on several benchmark datasets and compare FMOR with state-of-the-art approaches. The results show that FMOR noticeably outperforms all counterparts, especially in case of sparse feature space.

The rest of this paper is organized as follows. Section 2 briefly discusses the literature of ordinal regression. Section 3 gives the details of the proposed factorization machines for ordinal regression. Section 4 reports the experimental results. Section 5 gives some conclusive remarks.

2 Related Work

Generally, ordinal regression methods can be classified into three categories: (i) naive methods, (ii) ordinal binary decomposition methods, and (iii) threshold-based methods.

Naive Methods. Ordinal regression, akin to nominal classification and metric regression, can be simplified into these conventional supervised learning paradigms by either ignoring the ordinal relationship among classes or casting ordinal labels into real values. A more advanced method of this type is to transform ordinal regression as cost-sensitive classification, in which the ordinal information is encoded as misclassification costs [10].

Ordinal Binary Decomposition Methods. The main idea of ordinal binary decomposition methods is to decompose the ordinal classes into several binary pairs, each modeled by single or multiple traditional classifiers. Lin et al. [12] proposed a reduction framework from ordinal regression to binary classification: each sample is extended with a series of ordinal patterns, then a binary classifier is learned for each ordinal class that answers the question: "Is the rank of $\mathbf{x}$ greater than r or not?". Liu et al. [14] made use of triplets with each element from a different rank as samples and a binary classifier is learned for each ordinal class that answers the question: "Is the rank of $\mathbf{x}$ greater than $r-1$ and smaller than $r+1$?".

Threshold-Based Methods. Threshold-based methods have been one popular technique for handling ordinal samples. Threshold-based methods aim to estimate: (i) a latent regression function $f(\mathbf{x})$ that maps the input feature space to a one-dimensional real space; and (ii) a set of thresholds $b_1 \leq \cdots \leq b_R$ that cast the real-valued $f(\mathbf{x})$ into an interval corresponding to an ordinal class. The proportional odds model (POM) [15] is one of the first threshold-based methods and inspires many subsequent studies. Another well-known threshold-based ordinal regression method is Support Vector Ordinal Regression (SVOR) [3,20] that generalize the "large margin" principle adopted by support vector machines to ordinal regression. Two solutions to SVOR have been developed: one maximizes the margin of the closest neighboring classes (called fixed-margin strategy) and one maximizes the sum of margins of all classes (called sum-of-margin strategy). In a recent survey on ordinal regression [5], SVOR is proved to be the best threshold-based methods for its competitive performance on both prediction accuracy and training time.

3 The Proposed Method

In this section, we first give a preliminary introduction to the traditional FM, and then elaborate our proposed FMOR method.

3.1 Preliminary

Factorization Machines (FM) [19] are a generic model class that capable of dealing with high-dimensional and sparse features. Formally, FM takes as input a real valued vector $\mathbf{x} \in \mathbb{R}^d$, and estimates the target by modelling pairwise interactions of sparse features using low-rank latent factors. The model equation of FM is formulated as:

$$\hat{y}_{\mathrm{FM}}(\mathbf{x};\Theta) = w_0 + \sum_{j=1}^{d} w_j x_j + \sum_{j=1}^{d} \sum_{j'=j+1}^{d} \langle \mathbf{v}_j, \mathbf{v}_{j'} \rangle x_j x_{j'} \tag{1}$$

where the parameters Θ have to be estimated are:

$$w_0 \in \mathbb{R}; \quad \mathbf{w} \in \mathbb{R}^d; \quad \mathbf{V} = (\mathbf{v}_1, \cdots, \mathbf{v}_d) \in \mathbb{R}^{p \cdot d}$$

In Eq. 1, the first two items are linear combinations of every features with weights w_j $(1 \le j \le d)$ and a global bias w_0, and the last item is pairwise feature interactions using a factorized weighting schema $\hat{w}_{jj'} = \langle \mathbf{v}_j, \mathbf{v}_{j'} \rangle = \sum_{k=1}^{p} v_{jk} \cdot v_{j'k}$, where $\mathbf{v}_j$ is factor vector of the j-th feature, and $p \in \mathbb{N}^+$ is the hyper-parameter that defines the dimensionality of factor vectors. Feature factors in FM are commonly said to be low-rank, due to $p \ll d$. Compared with traditional ways (e.g., polynomial SVM) to model feature interactions using separated interaction weights, the factorization schema of FM can reduce the model complexity from $O(d^2)$ to $O(p \cdot d)$, which is a favored property for high-dimensional feature space.

Furthermore, FM is practically efficient for its linear computation time complexity. The model equation of FM in Eq. 1 can be reformulated as:

$$\hat{y}_{\mathrm{FM}}(\mathbf{x}; \Theta) = w_0 + \sum_{j=1}^{d} w_j x_j + \frac{1}{2} \sum_{k=1}^{p} \left(\left(\sum_{j=1}^{d} v_{jk} x_j \right)^2 - \sum_{j=1}^{d} v_{jk}^2 x_j^2 \right) \tag{2}$$

Equation 2 indicates that the model equation of FM has only linear time complexity in both d and p. In fact, the pairwise feature interaction can be only computed over the non-zero elements of $\mathbf{x}$, i.e., the computation complexity is $O(p \cdot N_z)$. Under sparsity settings, N_z can be much smaller than d, thus the computation of decision function of FM can be very efficient. In brief, FM provides a promising framework for handling high dimensional and sparse data.

3.2 Factorization Machines for Ordinal Regression

We realize Factorization Machines for Ordinal Regression (FMOR) by leveraging the threshold methodology. The basic idea is to introduce a set of consecutive thresholds to partition real line into several intervals which define the boundaries of ordinal classes.

Given an ordinal regression problem with R ordinal classes, OrdinalFM estimates the target of an input vector $\mathbf{x} \in \mathbb{R}^d$ as:

$$\hat{y}(\mathbf{x}) = \underset{r \in \{1, \cdots, R\}}{\arg\min} \{ f(\mathbf{x}) - b_r \le 0 \} \tag{3}$$

where $b_1, \cdots, b_R \in \mathbb{R}$ are the R thresholds partitioning real line into intervals, each corresponding to an ordinal class. Besides, the thresholds are required to satisfy the constraint $b_1 \le \cdots \le b_R$. For mathematical convenience, b_R is simply set as $+\infty$. $f(x)$ is the latent factor regression function that captures all possible interactions between features (up to second-order, practically):

$$f(\mathbf{x}) = \sum_{j=1}^{d} w_j x_j + \sum_{j=1}^{d} \sum_{j'=j+1}^{d} \langle \mathbf{v}_j, \mathbf{v}_{j'} \rangle x_j x_{j'} \tag{4}$$

Note that $f(\mathbf{x})$ is the same as the traditional FM model in Eq. 1 except the global bias item. In essence, FMOR extend the traditional FM by introducing a set of thresholds instead of the single global bias. The thresholds are used to map the regression function value $f(\mathbf{x})$ into ordinal targets. Particularly, an input is predicted as r if and only if $b_{r-1} < f(\mathbf{x}) \leq b_r$.

The model parameters of FMOR that have to be estimated are:

$$b_1, \cdots, b_{R-1} \in \mathbb{R}; \quad \mathbf{w} \in \mathbb{R}^d; \quad \mathbf{v}_1, \cdots, \mathbf{v}_d \in \mathbb{R}^p$$

Next, we discuss the learning procedure of FMOR, including the learning objective and the optimization algorithm.

The Learning Objective. Following the traditional supervised learning framework, the parameters Θ are learned from a given training set $\mathcal{D}$ that minimizes the following regularized empirical risk:

$$\mathcal{O}(\Theta, \mathcal{D}) = R_{\text{emp}}(\Theta, \mathcal{D}) + \lambda\Omega(\Theta) \tag{5}$$

where R_{emp} is the empirical risk of an ordinal regression model on the training data, and $\Omega(\cdot)$ is the regularization item. λ is the trade-off between the empirical risk and regularizer of model parameters.

In order to account for the ordinal relationship among targets when calculating the empirical risk, we consider measuring the predicting errors w.r.t. each ordinal class. Formally, give a training set $\mathcal{D} = \mathcal{D}^{(1)} \cup \cdots \cup \mathcal{D}^{(R)}$, where $\mathcal{D}^{(r)} = \{(\mathbf{x}_i, r), \cdots, (\mathbf{x}_{N_k}, r)\}$ $(1 \leq r \leq R)$ is the set of training samples with the class r, the empirical risk of a FMOR model is defined as:

$$R_{\text{emp}}(\Theta, \mathcal{D}) = \sum_{r=1}^{R} \sum_{i=1}^{N_r} \left(\sum_{k=1}^{r-1} \ell(f(\mathbf{x}) - b_k) + \sum_{k=r}^{R-1} \ell(b_k - f(\mathbf{x})) \right) \tag{6}$$

where $\ell(\cdot)$ is the surrogate loss function that penalizes an erroneous prediction. In fact, the empirical risk is contributed over all thresholds, including the lower-grading ones $(k = 1, \cdots, r-1)$ and the upper-grading ones $(k = r, \cdots, R)$, involved when a predicting error occurs.

Generally, a surrogate loss function is required to be monotonically decreasing in true positives. Moreover, smoothness is a derivable property such that efficient optimization techniques can be applied to estimate model parameters. Here, we adopt the smoothed hinge loss:

$$\ell(z) = \begin{cases} 0 & \text{if } z \geq 1 \\ \dfrac{(1-z)^2}{2} & \text{if } 0 < z < 1 \\ 0.5 - z & \text{if } z \leq 0 \end{cases} \tag{7}$$

As mentioned above, the threshold parameters need to satisfy the ordinal inequality constraint $b_1 \leq \cdots \leq b_R$. Interestingly, the constraint, although not being imposed on the learning procedure explicitly, can be automatically satisfied at the optimal solution of OrdinalFM, as will be shown in the following theorem.

Theorem 1. *Let $\Theta^* = (b_1^*, \cdots, b_{R-1}^*, \mathbf{w}^*, \mathbf{v}_1^*, \cdots, \mathbf{v}_d^*)$ be the optimal solution of the regularized empirical risk minimization problem in Eq. 5, i.e.,*

$$\Theta^* = \arg\min_{\Theta}\{R_{\text{emp}}(\Theta, \mathcal{D}) + \lambda\Omega(\Theta)\},$$

Then we have $b_1^* \leq \cdots \leq b_{R-1}^*$.

Theorem 1 not only establishes a nice property of FMOR but also induces a heuristic that is helpful for finding a better FMOR model. Theorem 1 leads to the following corollary.

Corollary 1. *Given two solution $\dot{\Theta} = (\dot{b}_1, \cdots, \dot{b}_{R-1}, \dot{\mathbf{w}}, \dot{\mathbf{v}}_1, \cdots, \dot{\mathbf{v}}_d)$ and $\ddot{\Theta} = (\ddot{b}_1, \cdots, \ddot{b}_{R-1}, \dot{\mathbf{w}}, \dot{\mathbf{v}}_1, \cdots, \dot{\mathbf{v}}_d)$, where $(\ddot{b}_1, \cdots, \ddot{b}_{R-1})$ is sorted in an ascending order of $(\dot{b}_1, \cdots, \dot{b}_{R-1})$, we have $R_{\text{emp}}(\ddot{\Theta}, \mathcal{D}) \leq R_{\text{emp}}(\dot{\Theta}, \mathcal{D})$.*

Due to space limitation, the proofs will be provided in the full version of the paper.

The Learning Algorithm. We employ the Adaptive Moment Estimation (Adam) [8] algorithm, a popular variant of stochastic gradient descent algorithm that uses adaptive per-parameter learning rates, to solve the regularized empirical risk minimization problem in Eq. 5. The main idea is to iterate over each sample $(\mathbf{x}, r)$ in the training set, and update model parameters towards the direction of negative gradient of the objective:

$$\theta^{(t)} = \theta^{(t-1)} - \eta^{(\theta,t)} \cdot \left(\frac{\partial R_{\text{emp}}(\Theta, \{(\mathbf{x}, r)\})}{\partial\theta} + \lambda\frac{\partial\Omega(\Theta)}{\partial\theta}\right) \tag{8}$$

where $\eta^{(\theta,t)}$ is the individual adaptive learning rate for θ at the t-th iteration.

For the empirical risk in Eq. 6, the gradient is given by

$$\frac{\partial R_{\text{emp}}(\Theta, \{(\mathbf{x}, r)\})}{\partial\theta} = \sum_{k=1}^{r-1}\frac{\partial\ell(f(\mathbf{x}) - b_k)}{\partial\theta} + \sum_{k=r}^{R-1}\frac{\partial\ell(b_k - f(\mathbf{x}))}{\partial\theta} \tag{9}$$

From Eq. 7, the gradient of the smoothed hinge loss is:

$$\frac{\partial\ell(z)}{\partial z} = \begin{cases} 0 & \text{if } z \geq 1 \\ z - 1 & \text{if } 0 < z < 1 \\ -1 & \text{if } z \leq 0 \end{cases} \tag{10}$$

From Eq. 4, the gradient of the factorized-based regression function is:

$$\begin{aligned} \frac{\partial f(\mathbf{x})}{\partial w_j} &= x_j \quad (j = 1, \cdots, d) \\ \frac{\partial f(\mathbf{x})}{\partial v_{j,l}} &= x_j \cdot \sum_{j' \neq j} v_{j',l}x_{j'} \quad (j = 1, \cdots, d; l = 1, \cdots, p) \end{aligned} \tag{11}$$

Through embedding Eqs. 10 and 11 into Eq. 9, we can obtain the gradient used in the optimization algorithm.

One thing to be noted here is gradient-based algorithms, though simple and efficient, are not guaranteed to find the global optimum solution, since the regularized empirical risk minimization problem is usually highly non-convex empirically. Thus the ordinal inequality constraint might be violated in the estimated parameters. Fortunately, according to Corollary 1, we can find better parameters, which not only satisfies the ordinal inequality constraint but also achieve lower regularized empirical risk, by sorting the learned threshold parameters in ascending order.

4 Empirical Study

In this section, we report the results of the empirical studies on the proposed FMOR using several benchmark datasets.

4.1 Experimental Settings

As the proposed FMOR is essentially a threshold-based method, we select several state-of-the-art threshold-based ordinal regression methods as baselines. We also compare FMOR against the traditional FM.

- ORBoost: The thresholded ensemble model for ordinal regression proposed by Lin and Li [11]. The two implementations, namely ORBoost-LR (Ordinal Regression Boosting with Left-Right margins) and ORBoost-All (Ordinal Regression Boosting with All margins), are used as the baselines.
- SVOR: The support vector formulation for ordinal regression proposed by Chu and Keerthi [3]. The two implementations, namely SVOREX (Support Vector Ordinal Regression with EXplicit constraints) and SVORIM (Support Vector Ordinal Regression with IMplicit constraints), are used as the baselines. Both methods are implemented with Gaussian kernel (with kernel width as 1) and linear kernel, respectively.
- POMNN: The ordinal neural network based on the proportional odds model proposed by Gutiérrez [6].
- FM: The original factorization machines proposed by Rendle [19]. The FM model is learned with the regression least-squares loss and the predictions are rounded to the nearest ordinal class.

The ORBoost methods, the SVOR methods and the traditional FM are run using the publicly available implementations provided by the authors[1]. We implemented the POMNN method using TensorFlow. As for the proposed FMOR, we implemented it on basis of LibFM. The hyper-parameters of each method are chosen from a certain range (shown in Table 1) using 5-fold cross-validation within the training set. For other parameters of each method, we use default settings provided by the implementations.

[1] http://www.work.caltech.edu/~htlin/program/orensemble/.
http://www.gatsby.ucl.ac.uk/~chuwei/svor.htm.
http://www.libfm.org/.

Table 1. The ranges for hyper-parameter selection

Method	Hyper-parameter	Range
SVOR	Trade-off parameter	{0,0.001,0.01,0.1,1,10}
POMNN	Neurons in the hidden layer	{10,25,50,75,100}
	Learning rate	{0.001,0.005,0.01,0.05,0.1}
FM/FMOR	Dimensionality of factor vector p	{5,10,15}
	Trade-off parameter λ	{0,0.001,0.01,0.1,1,10}
	The general learning rate η	{0.001,0.005,0.01,0.05,0.1}

All the methods are evaluated using the following measures:

1. *MZE*: The Mean Zero–one Error (MZE) is the fraction of incorrect predictions:

$$MZE = \frac{1}{N} \sum_{i=1}^{N} [\![\hat{y}(\mathbf{x}_i) \neq y_i]\!]$$

2. *MAE*: The Mean Absolute Error (MAE) is the average absolute deviation of the predictions from the ground-truth:

$$MAE = \frac{1}{N} \sum_{i=1}^{N} |\hat{y}(\mathbf{x}_i) - y_i|$$

4.2 Prediction Accuracy

In this experiment, we compared the proposed FMOR against the baselines on 9 benchmark datasets which are taken from public machine learning data repositories[2]. All these datasets are real ordinal datasets with a varying number of samples, features and classes. We preprocess each dataset by normalizing every numeric attributes into $[0, 1]$ and transforming every categorical attributes to binary forms with one-hot encoding (one feature per value). As for the *winequality* dataset, we generate one more preprocessed dataset by transforming all attributes, including both numeric ones and categorical ones, to binary forms. To be specific, each numeric attribute in the original dataset is discretized into pre-defined bins and then converted into one-hot vectors. This dataset, denoted as *winequality*$_{0/1}$ in Table 2, is of high sparsity as each sample is described by a 180-dimension binary feature vector. The characteristics of benchmark datasets are described in Table 2.

[2] https://archive.ics.uci.edu/ml/.
https://www.openml.org.

Table 2. Characteristics of the benchmark datasets

Datasets	#sample	#feature	#class	Class distribution
Balance-scale	625	20	3	288, 49, 288
Car-evaluation	1728	21	4	1210, 384, 69, 65
User-knowledge	258	5	4	24, 83, 88, 63
Eucalyptus	736	89	5	130, 105, 214, 107, 180
School-grades	648	124	14	16, 14, 35, 35, 97, 104, 72, 81, 63, 49, 36, 29, 15, 2
Lecturer-eval	1000	20	5	93, 280, 403, 197, 27
Social-worker	1000	31	4	32, 352, 399, 217
Turkiye-eval	5820	32	5	1902, 1028, 792, 1252, 846
Winequality	4898	11	7	20, 163, 1457, 2198, 880, 175, 5
Winequality$_{0/1}$	4898	180	7	20, 163, 1457, 2198, 880, 175, 5

Each dataset is randomly split 5 times into training and testing sets with ratio 2:1. The averaged MZE and MAE over 5 runs along with the standard deviations are reported in Tables 3 and 4 (best in bold), respectively. From the results, we can see the proposed FMOR beats all baselines 6 of 10 times in terms of both MZE and MAE. Among the datasets that the proposed FMOR performs best, the most significant improvement is obtained on the *winequality*$_{0/1}$ dataset of which the feature space is sparser than others. This indicates that the advantage of the proposed FMOR can be more significant as the level of sparsity gets higher. Actually, in these datasets (i.e., *user-knowledge*, *eucalyptus* and *turkiye-eval*) that FMOR or traditional FM fails to outperform traditional ordinal regression methods, the sparsity issue rarely occurs. Taking the *user-knowledge* dataset[3] as an example, the attributes are all numerical ones such as the exam performance or the study time of a student. However, this result does not necessarily mean that the proposed FMOR cannot be applied to the ordinal regression problems with dense feature space. In fact, there is only a small gap between FMOR and the best-performed baselines on these datasets. Also note that the proposed FMOR still achieves the best performance on the dense dataset *winequality*. Among all the baselines, SVOR methods perform best in most cases. We also notice that POMNN does not perform as well as expected. We argue that more advanced techniques for training deep neural networks need to be employed to learn a better neural networks model for ordinal regression.

[3] https://archive.ics.uci.edu/ml/datasets/User+Knowledge+Modeling.

Table 3. The MZE results (means and standard deviations over 5 runs) on benchmark datasets

	Balance-scale	Car-evaluation	User-knowledge	Eucalyptus
ORBoost-LR	$0.0208_{\pm 0.012}$	$0.0230_{\pm 0.005}$	$0.1500_{\pm 0.031}$	$0.3696_{\pm 0.028}$
ORBoost-All	$0.0211_{\pm 0.013}$	$0.0249_{\pm 0.004}$	$0.1538_{\pm 0.009}$	$0.3634_{\pm 0.034}$
SVOREX-Gau	$0.0221_{\pm 0.013}$	$0.0191_{\pm 0.003}$	$0.1537_{\pm 0.019}$	$\mathbf{0.3452}_{\pm 0.041}$
SVOREX-Lin	$0.0187_{\pm 0.016}$	$0.0262_{\pm 0.006}$	$0.1431_{\pm 0.024}$	$0.3724_{\pm 0.016}$
SVORIM-Gau	$0.0225_{\pm 0.017}$	$0.0196_{\pm 0.003}$	$\mathbf{0.1362}_{\pm 0.014}$	$0.3655_{\pm 0.029}$
SVORIM-Lin	$\mathbf{0.0185}_{\pm 0.016}$	$0.0242_{\pm 0.004}$	$0.1430_{\pm 0.032}$	$0.3920_{\pm 0.044}$
POMNN	$0.0227_{\pm 0.014}$	$0.0258_{\pm 0.037}$	$0.1501_{\pm 0.018}$	$0.3493_{\pm 0.024}$
FM	$0.0193_{\pm 0.014}$	$0.0243_{\pm 0.019}$	$0.2418_{\pm 0.019}$	$0.4029_{\pm 0.016}$
FMOR	$0.0189_{\pm 0.012}$	$\mathbf{0.0174}_{\pm 0.002}$	$0.1436_{\pm 0.005}$	$0.3643_{\pm 0.026}$
	School-grades	Lecturer-eval	Social-work-decs	Turkiye-eval
ORBoost-LR	$0.8742_{\pm 0.024}$	$0.4030_{\pm 0.004}$	$0.4250_{\pm 0.016}$	$0.8210_{\pm 0.014}$
ORBoost-All	$0.8395_{\pm 0.030}$	$0.4023_{\pm 0.003}$	$0.4261_{\pm 0.014}$	$0.6187_{\pm 0.016}$
SVOREX-Gau	$0.8395_{\pm 0.030}$	$0.3780_{\pm 0.020}$	$0.4260_{\pm 0.013}$	$\mathbf{0.5878}_{\pm 0.012}$
SVOREX-Lin	$0.8332_{\pm 0.030}$	$0.3960_{\pm 0.015}$	$0.4250_{\pm 0.015}$	$0.5885_{\pm 0.013}$
SVORIM-Gau	$0.8240_{\pm 0.027}$	$0.3781_{\pm 0.002}$	$0.4160_{\pm 0.011}$	$0.7833_{\pm 0.012}$
SVORIM-Lin	$0.8257_{\pm 0.041}$	$0.4050_{\pm 0.005}$	$0.4200_{\pm 0.105}$	$0.6361_{\pm 0.035}$
POMNN	$0.8370_{\pm 0.020}$	$0.3671_{\pm 0.003}$	$0.4181_{\pm 0.013}$	$0.6319_{\pm 0.020}$
FM	$0.8381_{\pm 0.085}$	$0.3660_{\pm 0.017}$	$0.4191_{\pm 0.021}$	$0.6732_{\pm 0.028}$
FMOR	$\mathbf{0.7735}_{\pm 0.043}$	$\mathbf{0.3583}_{\pm 0.017}$	$\mathbf{0.3950}_{\pm 0.013}$	
	Winequality	Winequality$_{0/1}$		
ORBoost-LR	$0.5188_{\pm 0.012}$	$0.5194_{\pm 0.018}$		
ORBoost-All	$0.5360_{\pm 0.005}$	$0.5319_{\pm 0.018}$		
SVOREX-Gau	$0.4941_{\pm 0.005}$	$0.5153_{\pm 0.015}$		
SVOREX-Lin	$0.5329_{\pm 0.003}$	$0.5186_{\pm 0.011}$		
SVORIM-Gau	$0.5184_{\pm 0.009}$	$0.5273_{\pm 0.014}$		
SVORIM-Lin	$0.4984_{\pm 0.007}$	$0.5153_{\pm 0.018}$		
POMNN	$0.4816_{\pm 0.012}$	$0.5122_{\pm 0.011}$		
FM	$0.8989_{\pm 0.011}$	$0.5148_{\pm 0.021}$		
FMOR	$\mathbf{0.4839}_{\pm 0.018}$	$\mathbf{0.4767}_{\pm 0.011}$		

4.3 Training Efficiency

In this experiment, we evaluate the training efficiency of the proposed FMOR by comparing training time with other methods. In this experiment, we only consider SVOR methods for comparison for the consistent outperformance over other baselines. All comparison methods were run on a single core of an Intel(R) Xeon(R) CPU E7-4830 processor clocked at 2.13 GHz with access to 24 GB RAM.

Due to space limitation, only the results on the *turkiye-eval* dataset are reported. Figure 1 plots the training time of every comparison methods with varying sizes of the training dataset. It can be clearly seen that the proposed FMOR scales much better than SVOR methods. On large training sets with thousands of samples, training a FMOR model takes only seconds while training SVOR models can take several hours.

Table 4. The MAE results (means and standard deviations over 5-fold cross validation runs) on benchmark datasets

	Balance-scale	Car-evaluation	User-knowledge	Eucalyptus
ORBoost-LR	0.0307$_{\pm 0.019}$	0.0271$_{\pm 0.013}$	0.1660$_{\pm 0.038}$	0.3966$_{\pm 0.050}$
ORBoost-All	0.0310$_{\pm 0.021}$	0.0258$_{\pm 0.013}$	0.1702$_{\pm 0.011}$	0.3823$_{\pm 0.058}$
SVOREX-Gau	0.0309$_{\pm 0.019}$	0.0206$_{\pm 0.013}$	0.1701$_{\pm 0.023}$	**0.3768**$_{\pm 0.035}$
SVOREX-Lin	0.0275$_{\pm 0.022}$	0.0265$_{\pm 0.016}$	0.1583$_{\pm 0.029}$	0.4185$_{\pm 0.058}$
SVORIM-Gau	0.0317$_{\pm 0.024}$	0.0204$_{\pm 0.014}$	**0.1507**$_{\pm 0.017}$	0.3816$_{\pm 0.046}$
SVORIM-Lin	0.0283$_{\pm 0.024}$	0.0246$_{\pm 0.014}$	0.1583$_{\pm 0.039}$	0.4197$_{\pm 0.037}$
POMNN	0.0321$_{\pm 0.021}$	0.0272$_{\pm 0.020}$	0.1662$_{\pm 0.022}$	0.3776$_{\pm 0.051}$
FM	**0.0248**$_{\pm 0.024}$	0.0248$_{\pm 0.020}$	0.2676$_{\pm 0.024}$	0.4414$_{\pm 0.027}$
FMOR	0.0281$_{\pm 0.019}$	**0.0188**$_{\pm 0.010}$	0.1589$_{\pm 0.006}$	0.3784$_{\pm 0.037}$
	School-grades	Lecturer-eval	Social-work-decs	Turkiye-eval
ORBoost-LR	2.7298$_{\pm 0.031}$	0.4369$_{\pm 0.016}$	0.4470$_{\pm 0.020}$	1.3115$_{\pm 0.025}$
ORBoost-All	2.3175$_{\pm 0.058}$	0.4349$_{\pm 0.014}$	0.4511$_{\pm 0.018}$	1.2529$_{\pm 0.032}$
SVOREX-Gau	2.3175$_{\pm 0.058}$	0.4140$_{\pm 0.011}$	0.4360$_{\pm 0.017}$	1.1560$_{\pm 0.042}$
SVOREX-Lin	2.2938$_{\pm 0.033}$	0.4299$_{\pm 0.020}$	0.4380$_{\pm 0.060}$	1.1663$_{\pm 0.035}$
SVORIM-Gau	2.1029$_{\pm 0.100}$	0.4130$_{\pm 0.009}$	0.4290$_{\pm 0.012}$	1.0979$_{\pm 0.016}$
SVORIM-Lin	1.9158$_{\pm 0.056}$	0.4380$_{\pm 0.012}$	0.4330$_{\pm 0.052}$	1.0450$_{\pm 0.039}$
POMNN	2.0764$_{\pm 0.091}$	0.4020$_{\pm 0.013}$	0.4290$_{\pm 0.050}$	**1.0297**$_{\pm 0.035}$
FM	2.1595$_{\pm 0.035}$	0.4010$_{\pm 0.021}$	0.4431$_{\pm 0.020}$	1.3324$_{\pm 0.042}$
FMOR	**1.8970**$_{\pm 0.027}$	**0.3910**$_{\pm 0.019}$	**0.4100**$_{\pm 0.021}$	1.1998$_{\pm 0.057}$
	Winequality	Winequality$_{0/1}$		
ORBoost-LR	0.5847$_{\pm 0.024}$	0.5892$_{\pm 0.030}$		
ORBoost-All	0.6107$_{\pm 0.015}$	0.6058$_{\pm 0.015}$		
SVOREX-Gau	0.5545$_{\pm 0.015}$	0.5835$_{\pm 0.024}$		
SVOREX-Lin	0.6074$_{\pm 0.013}$	0.5863$_{\pm 0.016}$		
SVORIM-Gau	0.5870$_{\pm 0.016}$	0.5998$_{\pm 0.019}$		
SVORIM-Lin	0.5574$_{\pm 0.017}$	0.5820$_{\pm 0.015}$		
POMNN	0.5492$_{\pm 0.017}$	0.5820$_{\pm 0.016}$		
FM	2.0390$_{\pm 0.055}$	0.5940$_{\pm 0.406}$		
FMOR	**0.5384**$_{\pm 0.025}$	**0.5306**$_{\pm 0.015}$		

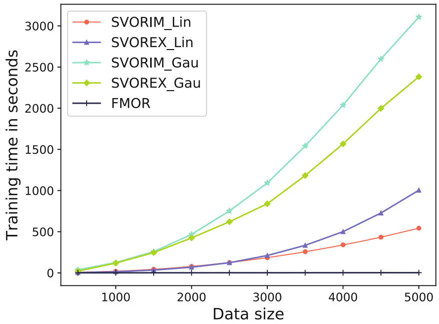

Fig. 1. Training time with varying dataset size

5 Conclusion

In this paper, we put forward Factorization Machines for Ordinal Regression (FMOR), a latent factor model addressing the sparsity issue in ordinal regression problems. Using the factorization machines as the base generic framework for modeling sparse feature space, we incorporate the threshold methodology to handle the ordinal targets in a proper way. We experimentally show that FMOR has the dual advantages of effectiveness and efficiency, and can be applied not only to sparse ordinal data, but competitive results can even be obtained for dense data. Future work includes applying FMOR to model ordinal user preference scores in recommender systems.

Acknowledgement. This work is partially supported by Natural Science Foundation of China (61602278, 71704096 and 31671588), Sci. & Tech. Development Fund of Shandong Province of China (ZR2017MF027), the Humanities and Social Science Research Project of the Ministry of Education (18YJAZH017), the Taishan Scholar Climbing Program of Shandong Province, and SDUST Research Fund (2015TDJH102).

References

1. Alp, A.: Structural shifts in credit rating standards. J. Financ. **68**(6), 2435–2470 (2013)
2. Beckham, C., Pal, C.: Unimodal probability distributions for deep ordinal classification, pp. 411–419 (2017)
3. Chu, W., Keerthi, S.S.: New approaches to support vector ordinal regression. In: Proceedings of the 22nd International Conference on Machine Learning, pp. 145–152 (2005)
4. Goh, C.K., Liu, Y., Kong, A.W.: A constrained deep neural network for ordinal regression. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 831–839 (2018)
5. Gutierrez, P.A., Perez-Ortiz, M., Sanchez-Monedero, J., Fernandez-Navarro, F., Hervas-Martinez, C.: Ordinal regression methods: survey and experimental study. IEEE Trans. Knowl. Data Eng. **28**(1), 127–146 (2016)

6. Gutiérrez, P.A., Tiňo, P., Hervás-Martínez, C.: Ordinal regression neural networks based on concentric hyperspheres. Neural Netw. **59**, 51–60 (2014)
7. Huang, X., Zhang, L., Wang, B., Zhang, Z., Li, F.: Feature weight estimation based on dynamic representation and neighbor sparse reconstruction. Pattern Recogn. **81**, 388–403 (2018)
8. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
9. Koren, Y., Sill, J.: Collaborative filtering on ordinal user feedback. In: Proceedings of the 23th International Joint Conference on Artificial Intelligence (2013)
10. Li, L., Lin, H.T.: Ordinal regression by extended binary classification. In: Advances in Neural Information Processing Systems, pp. 865–872 (2007)
11. Lin, H.T., Li, L.: Large-margin thresholded ensembles for ordinal regression: theory and practice. In: Proceedings of the International Conference on Algorithmic Learning Theory, pp. 319–333 (2006)
12. Lin, H.T., Li, L.: Reduction from cost-sensitive ordinal ranking to weighted binary classification. Neural Comput. **24**(5), 1329–1367 (2012)
13. Liu, X., Zou, Y., Song, Y., Yang, C., You, J., Kumar, B.V.K.V.: Ordinal regression with neuron stick-breaking for medical diagnosis. In: Leal-Taixé, L., Roth, S. (eds.) ECCV 2018. LNCS, vol. 11134, pp. 335–344. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-11024-6_23
14. Liu, Y., Kong, A.W.K., Goh, C.K.: Deep ordinal regression based on data relationship for small datasets. In: Proceedings of the 26th International Joint Conferences on Artificial Intelligence, pp. 2372–2378 (2017)
15. McCullagh, P.: Regression models for ordinal data. J. Roy. Stat. Soc.: Ser. B (Methodol.) **42**(2), 109–127 (1980)
16. Ni, W., Liu, T., Zeng, Q., Zhang, X., Duan, H., Xie, N.: Robust factorization machines for credit default prediction. In: Geng, X., Kang, B.-H. (eds.) PRICAI 2018. LNCS (LNAI), vol. 11012, pp. 941–953. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-97304-3_72
17. Pan, Z., Chen, E., Liu, Q., Xu, T., Ma, H., Lin, H.: Sparse factorization machines for click-through rate prediction. In: Proceedings of the IEEE 16th International Conference on Data Mining, pp. 400–409 (2016)
18. Qiang, R., Liang, F., Yang, J.: Exploiting ranking factorization machines for microblog retrieval. In: Proceedings of the 22nd ACM International Conference on Conference on Information & Knowledge Management, pp. 1783–1788 (2013)
19. Rendle, S.: Factorization machines with libFM. ACM Trans. Intell. Syst. Technol. **3**(3), 57 (2012)
20. Shashua, A., Levin, A.: Ranking with large margin principle: two approaches. In: Advances in Neural Information Processing Systems, pp. 961–968 (2003)
21. Sun, B.Y., Li, J., Wu, D.D., Zhang, X.M., Li, W.B.: Kernel discriminant learning for ordinal regression. IEEE Trans. Knowl. Data Eng. **22**(6), 906–910 (2010)
22. Tian, Q., Zhang, W., Wang, L., Chen, S., Yin, H.: Robust ordinal regression induced by lp-centroid. Neurocomputing **313**, 184–195 (2018)
23. Tran, T., Phung, D., Luo, W., Venkatesh, S.: Stabilized sparse ordinal regression for medical risk stratification. Knowl. Inf. Syst. **43**(3), 555–582 (2015)
24. Wang, H., Shi, Y., Niu, L., Tian, Y.: Nonparallel support vector ordinal regression. IEEE Trans. Cybern. **47**(10), 3306–3317 (2017)
25. Zhu, M., Aggarwal, C.C., Ma, S., Zhang, H., Huai, J.: Outlier detection in sparse data with factorization machines. In: Proceedings of the 2017 ACM Conference on Information and Knowledge Management, pp. 817–826 (2017)