Robust Factorization Machines for Credit Default Prediction

Weijian Ni[1], Tong Liu[1(✉)], Qingtian Zeng[1], Xianke Zhang[1], Hua Duan[1], and Nengfu Xie[2]

[1] College of Computer Science and Engineering, Shandong University of Science and Technology, Qingdao, China
niweijian@gmail.com, liu_tongtong@foxmail.com, qt_zeng@163.com, zhangxianke@sdust.edu.cn, huaduan59@163.com

[2] Agricultural Information Institute, Chinese Academy of Agricultural Sciences, Beijing, China
xienengfu@caas.cn

Abstract. Credit default prediction is a topic of great importance in lending industry. Just like many real-world applications, the dataset in the task is often class-imbalanced and noisy, degrading the performance of most machine learning methods. In this paper, we propose an extension of Factorization Machines, named RobustFM, to address the problem of class-imbalance and noisiness in the credit default prediction task. The proposed RobustFM employs a smoothed asymmetric Ramp loss function, into which truncation and hinge parameters are introduced to facilitate noise tolerance and imbalanced learning. Experimental results on several real credit datasets show that RobustFM significantly outperforms state-of-the-art methods in terms of F-measure.

Keywords: Factorization machines · Ramp loss
Imbalanced classification · Credit default prediction

1 Introduction

Credit default, generally refers to failure to pay interest or principal on a loan when due, is a primary potential source of risk in lending business. Distinguishing credit applicants with high default risk from credit-worthy ones has been identified as a crucial issue for risk management in financial institutions. Over the last decades, researchers and practitioners have sought to develop credit models using modern machine learning techniques [1] for their ability to model complex multivariate functions without rigorous assumptions for the input data.

Despite encouraging successes in recent studies, accurate predictive analysis of credit default through using machine learning techniques is by no means a trivial task. Many of the challenges stem from the fact that the data in the task, i.e., samples of credit applicants, is generally imbalanced and noisy. Defaults, which are often of more focused interests in the credit default prediction task,

X. Geng and B.-H. Kang (Eds.): PRICAI 2018, LNAI 11012, pp. 941–953, 2018.
https://doi.org/10.1007/978-3-319-97304-3_72

would only hit a small segment of credit customers in a real credit business [2]. This results in a heavily skewed class distribution of credit data. In addition, credit data is collected from loan records of financial institutions; however, due to privacy issues, system malfunctions or even human error, historical loan records are often incomplete or erroneous, making the credit data noisy. Therefore, in order to facilitate responsible decision-making for credit granting, the problems of class-imbalance and noisiness in credit data should be fully addressed.

Factorization machines (FM), proposed by Rendle [3] in the context of recommendation system, is a novel predictive model that maps a number of predictor variables to some target. The advantages FM offers over traditional classification approaches is that it provides a principled way to model second-order (up to arbitrary order in theory) variable interactions in linear complexity. FM has shown great promise in a number of prediction tasks, such as context-aware recommendation [4,5] and click-through rate prediction [6–8]. However, the potential of exploiting FM in credit risk evaluation has been little investigated so far. We argue that FM is powerful in credit default prediction task for at least the following reasons. First, for the task of credit default prediction, the combinations of predictor variables (e.g., family, age, and salary), usually much more discriminative than single ones, can be naturally modeled through variable interactions in FM. Second, FM embeds features into a low-rank latent space such that variable interactions can be estimated under high sparsity; thus FM can be viewed as a favored formulism for tackling sparse credit data.

In this work, we explore the use of FM for credit default prediction, with an emphasis on the class-imbalanced and noisy natural of credit data. We incorporate a new non-convex loss function into the learning process of FM and give rise to a novel **Robust F**actorization **M**achines (RobustFM) model that enhances FM for prediction under class-imbalance and noisiness settings. The new non-convex loss function is essentially a smoothed asymmetric Ramp loss [9] with additional degrees of freedom to tolerate the noise and imbalanced class distribution of credit data. Unlike convex loss functions used in traditional FM, the new loss function is upper bounded so as to enhance the robustness of the learning procedure. Furthermore, asymmetric margins are introduced to push learning towards achieving a larger margin on the rare class (defaulters).

The rest of the paper is organized as follows. In Sect. 2, we present preliminaries of this work, including Credit Default Prediction and Factorization Machines. We then present the details of the proposed RobustFM in Sect. 3. Experiment results are shown in Sect. 4. Finally, we review related work and conclude the paper in Sects. 5 and 6, respectively.

2 Preliminary

2.1 Credit Default Prediction

In point view of machine learning, the credit default prediction task is generally formalized as binary classification. Formally, each credit applicant is represented by a set of features (e.g., applicant's age, monthly income, education,

employment and loan purpose), denoted as $\mathbf{x} \in \mathbb{R}^d$, where d is the number of features. Each credit applicant belongs to either of the two classes with a label $y \in \{+1, -1\}$. In this work, we use $+1$ and -1 to denote credit applicants with high default risk (hereafter, bad applicants) and low default risk (hereafter, good applicants), respectively.

Given a training set $D = D^{(+)} \cup D^{(-)} \in \mathbb{R}^d \times \{+1, -1\}$, in which $D^{(+)}$ and $D^{(-)}$ denote a set of historical bad and good credit applicants, respectively. In general, $|D^{(+)}| < |D^{(-)}|$. The goal of credit default prediction is to learn a function $f : \mathbb{R}^d \mapsto \{+1, -1\}$, which is capable of classifying a new credit applicant into one of the two classes.

2.2 Factorization Machines

Factorization Machines (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:

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

where the parameters Θ have to be estimated are:

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

In Eq. 1, the first two items on the right-hand-side are linear combinations of each features with weights w_j $(1 \leq j \leq d)$ and global bias w_0, and the last item on the right-hand-side 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. Feature factors in FM are commonly said to be low-rank, due to $p \ll d$.

In addition to theoretical soundness of low-rank feature factorization, FM is also practically efficient for its linear prediction time complexity. Computing pairwise feature interaction directly requires time complexity of $O(d^2)$; however, it has been shown that the pairwise feature interaction in FM can be computed in $O(pd)$ using the equivalent formulation of Eq. 1 [3]:

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

The model parameters Θ of FM can be estimated through minimizing empirical risk over training set D, together with regularization of parameters:

$$\mathcal{O}_D(\Theta) = \frac{1}{|D|} \sum_{(\mathbf{x}, y) \in D} \ell\left(y, \hat{y}(\mathbf{x}; \Theta)\right) + \sum_{\theta \in \Theta} \lambda_\theta \theta^2 \tag{3}$$

where $\ell(y, \hat{y})$ is the loss function to evaluate the disagreement of the prediction value $\hat{y}$ with the actual label y. Without confusion, we sometimes use $\hat{y}$ to represent the prediction $\hat{y}(\mathbf{x}; \Theta)$ in the rest of the paper.

For binary classification, the most widely adopted loss function in FM is Logistic loss:

$$\ell_{\text{logit}}(y, \hat{y}) = \ln\left(1 + e^{-y\hat{y}}\right)$$

Despite effectiveness in various prediction tasks, FM still suffers from the curse of learning from imbalanced and noisy data. When the one class vastly outnumbers others, the learning objective in Eq. 3 can be dominated by instances from the major class. As such, FM tends to be overwhelmed by the major class, ignoring the minor, yet important ones, which is bad applicants in credit default prediction task. Furthermore, the Logistic loss, as convex loss functions, gives high penalties to those misclassified samples far from the origin, increasing the chances of outliers having a considerable contribution to the global loss. Thus the parameters estimated through optimizing Eq. 3 may be inevitably biased by outliers in noisy datasets, leading to a suboptimal predictive model that attempts to account for these outliers.

In this work, instead of Logistic loss, we incorporate into FM a new smoothed asymmetric Ramp loss allowing for class-dependent and up-bounded penalties for misclassified instances. This results in *RobustFM*, a new extension of FM that addresses imbalanced and noisy class distribution simultaneously, greatly improving the accuracy of credit default prediction under real-world scenarios.

3 Smooth Asymmetric Ramp Loss

Ramp loss, a non-convex loss function proposed by Collobert [9], is essentially a "truncated" version of Hinge loss used in support vector machines:

$$\ell_{\text{R}}(y, \hat{y}; \gamma) = \begin{cases} 1 - \gamma & \text{if } y\hat{y} < \gamma \\ 1 - y\hat{y} & \text{if } \gamma \leq y\hat{y} \leq 1 \\ 0 & \text{if } y\hat{y} > 1 \end{cases}$$

Intuitively, Ramp loss is constructed by flattening Hinge loss when the so-called functional margin $y\hat{y}$ smaller than a predefined parameter $\gamma < 0$. In other words, a fixed non-zero penalty $1 - \gamma$, rather than linear penalty $1 - y\hat{y}$ in Hinge loss, is applied to the samples mistakenly predicted far away from the origin (i.e., $y\hat{y} < \gamma$).

Studies have proven Ramp loss's superiority over Hinge loss in terms of robustness to noisy labels [10,11]. However, Ramp loss applies a unified penalty, either $1-\gamma$ or $1-y\hat{y}$, to all samples no matter to which class they belong. Similar as Logistic loss and Hinge loss, the empirical risk based on Ramp loss will be dominated by negative instances if the class distribution is highly imbalanced. Ramp loss thus still suffering from imbalanced class distribution.

One way to address the class-imbalance problem is to apply class-dependent penalties to the training errors. We introduce new parameters in Ramp loss to

control the degree of penalty for positive and negative classes, and construct an **asymmetric Ramp (aRamp) loss**:

$$\ell_{\mathrm{aR}}(y,\hat{y};\gamma,\tau^{(+)},\tau^{(-)}) = \begin{cases} \tau^{(y)} - \gamma & \text{if } y\hat{y} < \gamma \\ \tau^{(y)} - y\hat{y} & \text{if } \gamma \le y\hat{y} \le \tau^{(y)} \\ 0 & \text{if } y\hat{y} > \tau^{(y)} \end{cases} \quad (y \in \{+1,-1\})$$

There are three parameters in asymmetric Ramp loss: γ ($\gamma < 0$) is truncation parameter that decides the point to flatten the loss function; $\tau^{(+)}$ and $\tau^{(-)}$ are the hinge parameters for false negative and false positive errors, respectively. In general, $\tau^{(+)} > \tau^{(-)} \ge 1$, since false negative error is considered more serious than false positive error in imbalanced classification problems.

One should note that the asymmetric Ramp loss is not differentiable at the truncation point ($y\hat{y} = \gamma$) and the hinge points ($y\hat{y} = \tau^{(y)}$), whereas smoothness is a desired property for gradient-based optimization techniques, e.g., stochastic gradient descent and alternating coordinate descent, which have been widely used for training FM. Motivated by the smoothing mechanism adopted in designing Huber loss [12], we make use of smooth quadratic function to approximate the asymmetric Ramp loss at the non-smooth points. More specifically, we derive a **smooth asymmetric Ramp (saRamp) loss** as follows:

$$\ell_{\mathrm{saR}}(y,\hat{y};\gamma,\tau^{(+)},\tau^{(-)},\delta) = \begin{cases} \tau^{(y)} - \gamma & \text{if } y\hat{y} < \gamma - \delta \\ \tau^{(y)} - y\hat{y} - \dfrac{(\gamma+\delta-y\hat{y})^2}{4\delta} & \text{if } \gamma-\delta \le y\hat{y} \le \gamma+\delta \\ \tau^{(y)} - y\hat{y} & \text{if } \gamma+\delta < y\hat{y} < \tau^{(y)} - \delta \\ \dfrac{(\tau^{(y)}+\delta-y\hat{y})^2}{4\delta} & \text{if } \tau^{(y)}-\delta \le y\hat{y} \le \tau^{(y)}+\delta \\ 0 & \text{if } y\hat{y} > \tau^{(y)} + \delta \end{cases}$$

The saRamp loss is quadratic for small interval around the truncation point $[\gamma-\delta, \gamma+\delta]$ and the hinge point $[\tau^{(y)}-\delta, \tau^{(y)}+\delta]$, and linear for other values. Figure 1 illustrates the aRamp loss and saRamp loss with different interval length δ. It is easy to verify that $\lim_{\delta\to 0} \ell_{\mathrm{saR}}(y,\hat{y};\gamma,\tau^{(+)},\tau^{(-)},\delta) = \ell_{\mathrm{aR}}(y,\hat{y};\gamma,\tau^{(+)},\tau^{(-)})$. We omit the proof due to brevity. In practice, we set $\delta = 0.1$. Without ambiguity, we briefly denote saRamp loss $\ell_{\mathrm{saR}}(y,\hat{y};\gamma,\tau^{(+)},\tau^{(-)},\delta)$ as $\ell_{\mathrm{saR}}(y,\hat{y})$ hereafter.

The derivative of the saRamp loss w.r.t. the functional margin can be easily derived as follows:

$$\frac{\partial \ell_{\mathrm{saR}}(y,\hat{y})}{\partial(y\hat{y})} = \begin{cases} 0 & \text{if } y\hat{y} < \gamma - \delta \\ \dfrac{\gamma+\delta-y\hat{y}}{2\delta} - 1 & \text{if } \gamma-\delta \le y\hat{y} \le \gamma+\delta \\ -1 & \text{if } \gamma+\delta < y\hat{y} < \tau^{(y)} - \delta \\ -\dfrac{\tau^{(y)}+\delta-y\hat{y}}{2\delta} & \text{if } \tau^{(y)}-\delta \le y\hat{y} \le \tau^{(y)}+\delta \\ 0 & \text{if } y\hat{y} > \tau^{(y)} + \delta \end{cases} \tag{4}$$

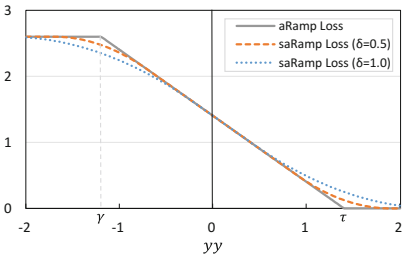


Fig. 1. Asymmetric Ramp loss and smooth asymmetric Ramp loss

4 Parameter Estimation

To solve the highly non-convex problem in Eq. 3 in a large scale, iterative optimization methods are usually preferred, due to the simplicity nature and flexibility in the choices of loss function. In this work, we employ Stochastic Gradient Descent (SGD), one of the most popular optimization method in factorization models, to estimate the parameters of RobustFM. Simply put, SGD updates parameters iteratively until convergence. In each iteration, an instance $(\mathbf{x}, y)$ is randomly drawn for training data D, and the update is performed towards the direction of negative gradient of the objective w.r.t. each parameter $\theta \in \Theta$:

$$\theta^{(t)} = \theta^{(t-1)} - \eta \cdot \left(\frac{\partial \mathcal{O}_{\{(\mathbf{x},y)\}}(\Theta^{(t-1)})}{\partial \theta} \right) \tag{5}$$

where $\eta > 0$ is the learning rate of gradient descendent.

Plugging the learning objective Eq. 3 into Eq. 5, we derive the parameter updating formula:

$$\theta^{(t)} = \theta^{(t-1)} - \eta \cdot \left(\frac{\partial \ell_{\text{saR}}(y, \hat{y}(\mathbf{x}; \Theta^{(t-1)}))}{\partial \theta} + 2\lambda_\theta \theta^{(t-1)} \right)$$

Applying the chain rule to Eq. 4 yields the derivative of the saRamp loss w.r.t. model parameters:

$$\begin{aligned} \frac{\partial \ell_{\text{saR}}(y, \hat{y})}{\partial \theta} &= \frac{\partial \ell_{\text{saR}}(y, \hat{y})}{\partial (y\hat{y})} \cdot \frac{\partial (y\hat{y})}{\partial \theta} \\ &= \begin{cases} y \cdot \left(\frac{\gamma + \delta - y\hat{y}}{2\delta} - 1 \right) \cdot \frac{\partial \hat{y}}{\partial \theta} & \text{If } \gamma - \delta \leq y\hat{y} \leq \gamma + \delta \\ -y \cdot \frac{\partial \hat{y}}{\partial \theta} & \text{If } \gamma + \delta < y\hat{y} < \tau^{(y)} - \delta \\ -y \cdot \frac{\tau^{(y)} + \delta - y\hat{y}}{2\delta} \cdot \frac{\partial \hat{y}}{\partial \theta} & \text{If } \tau^{(y)} - \delta \leq y\hat{y} \leq \tau^{(y)} + \delta \\ 0 & \text{Otherwise} \end{cases} \end{aligned}$$

where $\frac{\partial \hat{y}}{\partial \theta}$ is the partial derivatives of model equation of FM w.r.t. each parameters. According to Eq. 2, it can be written as follows:

$$\frac{\partial \hat{y}}{\partial w_0} = 1$$
$$\frac{\partial \hat{y}}{\partial w_j} = x_j \ (1 \le j \le d)$$
$$\frac{\partial \hat{y}}{\partial v_{jk}} = x_j \sum_{j' \ne j} v_{j'k} x_{j'} \ (1 \le j \le d, 1 \le k \le p)$$

Given the above equations, the parameter estimation procedure for RobustFM is summarized in Algorithm 1. Note that, for each instance, the runtime complexity of Algorithm 1 remains the same as traditional FM, i.e., $O(p \cdot N_0(\mathbf{x}))$ where $N_0(\mathbf{x})$ denotes the number of non-zero features of the instance. Even so, the learning procedure of RobustFM is more computational efficient than that of traditional FM, because Algorithm 1 only iterates over the instances with non-zero gradient (line 5) whereas all instances in training data are to be handled in each iterations of traditional FM learning procedure.

Algorithm 1. ParaEstimate

Input: Training set D
Output: Model parameters Θ
1: Initial model parameters: $w_0 \leftarrow 0$; $\mathbf{w} \leftarrow (0, \cdots, 0)$; $\mathbf{V} \sim \mathcal{N}(0, 0.1)$;
2: **repeat**
3: **for** $(\mathbf{x}, y) \in D$ **do**
4: Predict current instance as $\hat{y}(\mathbf{x}; \Theta)$
5: **if** $\gamma - \delta < y \cdot \hat{y}(\mathbf{x}; \Theta) < \tau^{(y)} + \delta$ **then**
6: Update w_0, $\mathbf{w}$ and $\mathbf{V}$ according to Eq. 5.
7: **end if**
8: **end for**
9: **until** convergence

5 Experiments

5.1 Experimental Settings

Datasets. Several real-world credit datasets, including four public datasets and one private dataset, are used for empirical evaluation of the proposed RobustFM. A summary of the five datasets is illustrated in Table 1.

Australian, *German* and *Taiwan* are public credit datasets available from UCI Machine Learning Repository that have been widely used in the literature. *SomeCredit* is the dataset of Kaggle competition *Give Me Some Credit* that aims to predict the probability of future financial distress of loan borrowers. Besides these public datasets, we used a private dataset, denoted as *SD-RCB*, in this experiment. The dataset is sourced from a regional bank of China which provides micro-credit services to self-employed workers and farmer households.

In total, more than 50 thousands historical credit records are collected from the credit scoring system of the bank. The attributes of each credit records include custom demographics, credit application information, historical repayment behavior, and etc.

From Table 1, it has to be noted that the number of defaulters is always less than that of non-defaulters in all these datasets, and the default rate is even less than 10% in *SomeCredit* and *SD-RCB*, the two large-scale real-world credit datasets. This provides practical evidence of class-imbalance problem in real-world credit default prediction tasks.

Table 1. Statistics of credit datasets

Dataset	#samples	#attributes	Default rate (%)
Australian[a]	690	14	44.5
German[b]	1,000	20	30.0
Taiwan[c]	30,000	23	22.1
SomeCredit[d]	150,000	10	6.7
SD-RCB	54,893	46	9.2

[a]http://archive.ics.uci.edu/ml/datasets/statlog+(australian+credit+approval)
[b]https://archive.ics.uci.edu/ml/datasets/statlog+(german+credit+data)
[c]https://archive.ics.uci.edu/ml/datasets/default+of+credit+card+clients
[d]https://www.kaggle.com/c/GiveMeSomeCredit

Evaluation Measures. In order to compare the performance among different approaches, we employ the following three types of measures in this experiment:

- Accuracy (Acc). Accuracy aims to evaluate the correctness of categorical predictions: $Acc = \frac{1}{N}\sum_{i=1}^{N} \mathbb{I}_{[y_i \neq \text{sgn}(\hat{y}_i)]}$
- Brier Score (BS). Most classifiers give probabilistic predictions $\hat{y} = \hat{p}(\pm 1|\mathbf{x})$, rather than category predictions $\hat{y} = \pm 1$, making Accuracy calculated in an unnatural way – a categorical prediction can be only inferred by assigning a manually-tuned threshold to raw predictions. Unlike Accuracy, Brier Score aims to evaluate the correctness of the raw probabilistic predictions: $BS = \frac{1}{N}\sum_{i=1}^{N}\left(y_i - \hat{p}(y_i|\mathbf{x}_i)\right)^2$
- Precision (Pre), Recall (Rec) and F-measure (F_1). One important shortage of Acc and BS is that a classifier is evaluated without taking into account the variations between classes. However, in the credit default prediction tasks, the correctness of predictions on defaulters is much more important than that on non-defaulters. We thus employ three performance measures: Precision, Recall and F-measure. Essentially, Precision and Recall measure the type-I error (non-defaulter classified as defaulter) and the type-II error (defaulter classified

as non-defaulter), respectively, and F-measure is the harmonic average of Precision and Recall.

Baselines. To verify the advantage of the proposed RobustFM, we compare it with several state-of-the-art credit default prediction models. First, the most widely used classification techniques in the task of credit default prediction [1], including logistic regression (LR), neural networks (NN) and support vector machines (SVM), are selected as baselines. Second, the traditional FM, as well as its extensions with traditional Hinge loss and Ramp loss, are applied to the task of credit default prediction and selected as baselines.

5.2 Performance Evaluation

In this experiment, five-fold cross validation is performed on each dataset and the average performance on the five folds is reported. Table 2 presents the experimental results and comparisons on each dataset.

It can be seen from Table 2 that the proposed RobustFM, compared to baseline methods, achieves the highest F_1 score on all the five datasets. We perform statistical significance test to check whether the improvements are significant. More specifically, paired t-tests is applied on the predicted results obtained by RobustFM and the nearest counterpart. The results indicate that the improvements of RobustFM are significant with p-value ≤ 0.05 on datasets *German* and *Taiwan* and p-value ≤ 0.01 on datasets *SomeCredit* and *SD-RCB*, marked with single and double asterisks in Table 2, respectively. In fact, the imbalanced ratio of the datasets *SomeCredit* and *SD-RCB* is much higher than that of others. The comparison results prove the effectiveness of RobustFM in dealing with imbalanced data, especially with high imbalanced ratio and large size.

From Table 2, we have the following more observations:

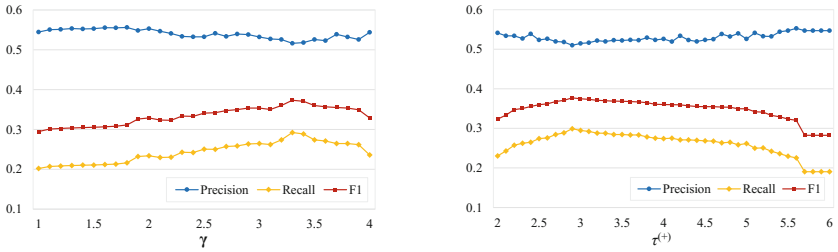
i. Besides F_1, RobustFM achieves the highest *Recall* on four of the five datasets. This is in fact favored in real-world credit default prediction tasks in which missing a true defaulter in predictions is typically perceived as a more severe error than misclassifying a non-defaulter as a defaulter.
ii. Achieving highest F_1 doesn't necessarily result in highest *Accuracy* and *BS*. As a matter of fact, RobustFM only achieves the highest *Accuracy* on dataset *Australian* which is a rather balanced dataset of small size. However, it is well recognized that *Accuracy* is not suitable for evaluating performances on class-imbalanced setting.
iii. Among all the baselines, FM-based methods (FM, FM_{Hinge}, and FM_{Ramp}) perform better than most traditional methods (LR, NN, and SVM) in terms of almost all performance measures. This result coincides with the findings of previous studies on FM from a variety of tasks such as click-through rate prediction and context-aware recommendation, and further verifies our intuition of the advantages of FM when applying to the credit prediction tasks described before.

Table 2. Prediction performance of each approaches

Dataset	Methods	$Acc(\%)$	BS	$Pre(\%)$	$Rec(\%)$	$F_1(\%)$
Australian	LR	68.84	0.2976	72.10	88.25	79.34
	NN	67.97	0.2443	**73.00**	83.76	78.00
	SVM	67.97	0.2488	68.14	99.15	80.77
	FM	70.69	0.2166	72.34	89.47	80.00
	FM-Hinge	70.68	0.2325	71.43	92.10	80.45
	FM-Ramp	72.41	0.2153	72.92	92.11	81.39
	RobustFM	**73.28**	**0.2114**	72.73	**94.74**	**82.29**
German	LR	**75.70**	**0.2902**	**62.18**	49.33	54.89
	NN	75.30	0.3004	60.56	51.67	55.63
	SVM	72.50	0.3464	53.70	59.67	56.52
	FM	74.85	0.3477	53.85	60.87	57.14
	FM-Hinge	71.86	0.4123	49.15	63.04	55.24
	FM-Ramp	74.25	0.4327	52.63	65.22	58.25
	RobustFM	70.66	0.3681	48.06	**80.43**	**60.17***
Taiwanese	LR	80.73	0.2688	63.51	30.76	41.38
	NN	81.06	0.2622	62.73	35.97	45.64
	SVM	79.73	0.2812	58.61	38.16	44.94
	FM	**81.96**	0.1668	**65.48**	35.67	46.18
	FM-Hinge	81.14	**0.1628**	58.60	44.61	50.65
	FM-Ramp	81.26	0.1932	59.18	43.96	50.45
	RobustFM	77.78	0.1932	48.95	**55.58**	**52.06***
SomeCredit	LR	93.59	0.2002	55.49	19.29	28.62
	NN	93.43	0.2004	52.18	18.30	26.99
	SVM	93.58	0.1645	**55.78**	17.43	26.52
	FM	93.76	0.1469	53.17	23.16	32.26
	FM-Hinge	93.67	**0.1238**	55.45	21.25	30.73
	FM-Ramp	**93.96**	0.1513	54.74	24.14	33.50
	RobustFM	93.24	0.1611	47.20	**34.21**	**39.67****
SD-RCB	LR	52.38	0.5850	**67.82**	15.81	23.55
	NN	55.09	0.4426	55.90	13.35	20.19
	SVM	67.86	0.4209	53.19	16.49	24.41
	FM	90.19	0.2989	37.37	20.73	26.67
	FM-Hinge	**91.37**	**0.1764**	49.71	24.09	32.45
	FM-Ramp	85.83	0.3373	26.85	**37.54**	31.31
	RobustFM	88.62	0.2561	34.07	34.45	**34.26****

5.3 Hyper-parameter Study

Compared with traditional FM, there are two additional hyper-parameters: γ and $\tau^{(+)}$[1]. Experiments are performed to study how the prediction performance of RobustFM is affected by these parameters. More specifically, by fixing one of the parameters, we vary the other one and record the prediction performance in terms of *Precision*, *Recall* and F_1 (see Fig. 2(a) and (b)). Due to space limitation, only the results on the dataset *SomeCredit* are reported, and the results on other datasets are similar.



(a) Varying γ by fixing $\tau^{(+)} = 2.9$ (b) Varying $\tau^{(+)}$ by fixing $\gamma = 3.3$

Fig. 2. The effects of hyper-parameters

To choose the truncation parameter γ, one may typically start from a number slightly smaller than 0 and then decrease γ to tune the level of learning insensitivity of RobustFM. From Fig. 2(a), it can be seen that the prediction performance of RobustFM varies slightly when $\gamma \in (2, 4)$ and the optimal F_1 score is achieved around $\gamma = 3.3$. When γ is getting smaller and smaller, the robustness of RobustFM is reduced, causing performance to degrade as depicted in Fig. 2(a). Similarly, to choose the margin parameter $\tau^{(+)}$, one may typically start from a number slightly larger than 1.0 and increase $\tau^{(+)}$. While $\tau^{(+)}$ is increasing, the classification hyperplane is moving towards to the major class. This process can be illustrated in Fig. 2 in which the *Recall* increases constantly as $\tau^{(+)}$ increases from 2.0 to 3.0. From Fig. 2(b), it also can be seen that the prediction performance of RobustFM varies slightly when $\tau^{(+)} \in (2, 5)$ and the optimal F_1 score is achieved around $\tau^{(+)} = 2.9$. Overall, these experiments indicate that the proposed RobustFM performs quite steadily with wide-range values of truncation parameter and margin parameter.

6 Related Work

Credit default prediction has long been a central concern of financial risk management research. More recently, emerging machine learning techniques, instead

[1] In practice, the negative margin parameter $\tau^{(-)}$ is usually set as 1, thus $\tau^{(+)}$ can be just viewed as the relative positive margin.

of simple statistical methods, have been widely applied in the literature. Extensive studies have already demonstrated that machine learning techniques outperform classical statistical methods on various credit risk evaluation tasks. Until recently, almost all of the popular machine learning algorithms, e.g., support vector machines [13,14], decision tree [15] and neural networks [16,17] have been employed to construct credit risk model. Recent studies show that ensemble method that integrates predictions of several individual classifiers is a promising approach for credit risk modeling. A number of ensemble strategies have been proposed to construct more powerful credit risk models [18–20].

The class-imbalance problem in credit data has drawn attention in the literature. Several experimental studies have shown that most machine learning algorithms (e.g., decision tree, neural networks, and etc) perform significantly worse on imbalanced credit datasets [21,22]. Recently, A few studies have tried to tackling the class-imbalance problem in credit data by developing specific feature selection and ensemble strategies [23].

7 Conclusion and Future Work

In this paper, we propose a novel approach *RobustFM* for the credit default prediction task. Compared with existing machine learning based credit risk models, the main advantage of RobustFM is to address the issues of class-imbalance and noisiness in the credit data simultaneously. We demonstrate RobustFM's effectiveness on credit default prediction task via experimental evaluations on several real credit application datasets. It can be concluded that the proposed RobustFM is a worthwhile choice for the credit default prediction task.

Several issues could be considered for future work. For example, there are additional hyper-parameters in RobustFM that need to be tuned to yield good predictions. Further study should be continued to apply automated machine learning techniques to derive the optimal hyper-parameters automatically.

Acknowledgement. This work is partially supported by Natural Science Foundation of China (61602278, 71704096, 61472229 and 31671588), Sci. & Tech. Development Fund of Shandong Province (2016ZDJS02A11, 2014GGX101035 and ZR2017MF027), the Taishan Scholar Climbing Program of Shandong Province, and SDUST Research Fund (2015TDJH102).

References

1. Lessmann, S., Baesens, B., Seow, H.-V., Thomas, L.C.: Benchmarking state-of-the-art classification algorithms for credit scoring: an update of research. Eur. J. Oper. Res. **247**(1), 124–136 (2015)
2. Pluto, K., Tasche, D.: Estimating probabilities of default for low default portfolios. In: Engelmann, B., Rauhmeier, R. (eds.) The Basel II Risk Parameters, pp. 75–101. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-16114-8_5
3. Rendle, S.: Factorization machines with libFM. ACM Trans. Intell. Syst. Technol. (TIST) **3**(3), 57 (2012)

4. Wang, S., Li, C., Zhao, K., Chen, H.: Learning to context-aware recommend with hierarchical factorization machines. Inf. Sci. **409**, 121–138 (2017)
5. Rendle, S., Gantner, Z., Freudenthaler, C., Schmidt-Thieme, L.: Fast context-aware recommendations with factorization machines. In: Proceedings of the 34th ACM SIGIR, pp. 635–644 (2011)
6. Juan, Y., Zhuang, Y., Chin, W.-S., Lin, C.-J.: Field-aware factorization machines for CTR prediction. In: Proceedings of the 10th ACM RecSys, pp. 43–50 (2016)
7. Pan, Z., Chen, E., Liu, Q., Xu, T., et al.: Sparse factorization machines for click-through rate prediction. In: Proceedings of the 16th IEEE ICDM, pp. 400–409 (2016)
8. Guo, H., Tang, R., Ye, Y., Li, Z., He, X.: DeepFM: a factorization-machine based neural network for CTR prediction. arXiv preprint arXiv:1703.04247
9. Collobert, R., Sinz, F., Weston, J., Bottou, L.: Trading convexity for scalability. In: Proceedings of the 23rd ICML, pp. 201–208 (2006)
10. Ghosh, A., Kumar, H., Sastry, P.: Robust loss functions under label noise for deep neural networks. In: Proceedings of the 31th AAAI, pp. 1919–1925 (2017)
11. Cevikalp, H., Franc, V.: Large-scale robust transductive support vector machines. Neurocomputing **235**, 199–209 (2017)
12. Zhang, T.: Solving large scale linear prediction problems using stochastic gradient descent algorithms. In: Proceedings of the 21th ICML, p. 116 (2004)
13. Hens, A.B., Tiwari, M.K.: Computational time reduction for credit scoring: an integrated approach based on support vector machine and stratified sampling method. Expert Syst. Appl. **39**(8), 6774–6781 (2012)
14. Danenas, P., Garsva, G.: Selection of support vector machines based classifiers for credit risk domain. Expert Syst. Appl. **42**(6), 3194–3204 (2015)
15. Nie, G., Rowe, W., Zhang, L., Tian, Y., Shi, Y.: Credit card churn forecasting by logistic regression and decision tree. Expert Syst. Appl. **38**(12), 15273–15285 (2011)
16. Lisboa, P.J., et al.: Partial logistic artificial neural network for competing risks regularized with automatic relevance determination. IEEE Trans. Neural Netw. **20**(9), 1403–1416 (2009)
17. Marcano-Cedeno, A., Marin-De-La-Barcena, A., Jiménez-Trillo, J., Pinuela, J., Andina, D.: Artificial metaplasticity neural network applied to credit scoring. Int. J. Neural Syst. **21**(04), 311–317 (2011)
18. Ala'raj, M., Abbod, M.F.: A new hybrid ensemble credit scoring model based on classifiers consensus system approach. Expert Syst. Appl. **64**, 36–55 (2016)
19. Xiao, H., Xiao, Z., Wang, Y.: Ensemble classification based on supervised clustering for credit scoring. Appl. Soft Comput. **43**, 73–86 (2016)
20. Xia, Y., Liu, C., Li, Y., Liu, N.: A boosted decision tree approach using bayesian hyper-parameter optimization for credit scoring. Expert Syst. Appl. **78**, 225–241 (2017)
21. Louzada, F., Ferreira-Silva, P.H., Diniz, C.A.: On the impact of disproportional samples in credit scoring models: an application to a Brazilian bank data. Expert Syst. Appl. **39**(9), 8071–8078 (2012)
22. Brown, I., Mues, C.: An experimental comparison of classification algorithms for imbalanced credit scoring data sets. Expert Syst. Appl. **39**(3), 3446–3453 (2012)
23. Sun, J., Lang, J., Fujita, H., Li, H.: Imbalanced enterprise credit evaluation with DTE-SBD: decision tree ensemble based on SMOTE and bagging with differentiated sampling rates. Inf. Sci. **425**, 76–91 (2018)