Robust and Effective Factorization Machines

Anonymous

Optimization Algorithm

The original objective for classsification task takes the form:

$$\min_{\mathbf{w}\in\mathbf{R}^d,\mathbf{Z}\in\mathbf{S}_+^{d\times d}} \sum_{i=1}^{n} e_i(\max\{\max(y_i(\mathbf{w}^\top\mathbf{x}_i + \langle\mathbf{Z},\mathbf{x}_i\mathbf{x}_i^\top\rangle),0)-\epsilon_1,0\})^2 + \frac{\alpha}{2}\|\mathbf{w}\|^2 + \sum_s \min\{\lambda_s^2,\epsilon_3\}, \quad (1)$$

$$e_i = \begin{cases} \frac{1}{2error}, & 0 < error \le \epsilon_2; \\ 0, & otherwise \end{cases}$$

where $error = \max(y_i(\mathbf{w}^\top\mathbf{x}_i + \langle\mathbf{Z},\mathbf{x}_i\mathbf{x}_i^\top\rangle),0)-\epsilon_1$
The subgradient with respect to $\mathbf{Z}$ is

$$\triangledown_{\mathbf{Z},I} = \sum_{i=1}^{b}\mathbf{x}_i\mathbf{x}_i^\top + \beta\mathbf{P}_M\mathbf{P}_M^\top\mathbf{Z} \quad (2)$$

To incrementally calculate the SVD of $\mathbf{Z}-\eta\triangledown_{\mathbf{Z},I}$. Let the symmetric and low rank matrix $\mathbf{Z}$ has rank k and its economy SVD is $\mathbf{Z}=\mathbf{P}_k\Sigma_k\mathbf{P}_k^\top$. As matrix $\triangledown_{\mathbf{Z},I}$ is symmetric and low rank, we can represent it as $\mathbf{A}\mathbf{A}^\top$.

$$\triangledown_{\mathbf{Z},I} = \mathbf{X}\mathbf{X}^\top + \beta\mathbf{P}_M\mathbf{P}_M^\top\mathbf{Z} \quad (3)$$

Experiments

In this section, we empirically investigate whether our proposed RobFM method can achieve better and robust performance compared to original factorization machine model with fixed rank on benchmark datasets.

Experimental Testbeds and Setup

We conduct our experiments on four public datasets. Table 1 gives a brief summary of these datasets. All the datasets are normalized to have zero mean and unit variance in each dimension. To make fair comparison, all the algorithms are conducted over 5 experimental runs of different random permutations. We apply hinge loss for training and evaluate the performance of our proposed methods for classification task by measuring accuracy and hinge loss. For parameter settings, we perform grid search to choose the best parameters for each algorithm on the training set.

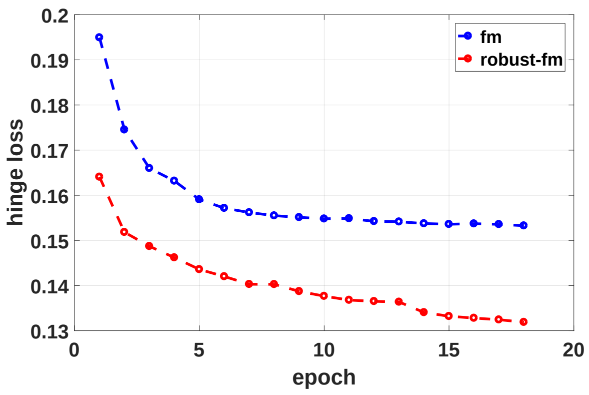
Dataset	#Training	#Test	#Feature	#class
phishing	7370	3685	68	2
w8a	49749	14951	300	2
protein	17766	6621	357	3
IJCNN	49990	91701	22	2
Covtype	387342	193670	54	2
MNIST	60000	10000	784	10

Table 1: Summary of datasets used in our experiments.

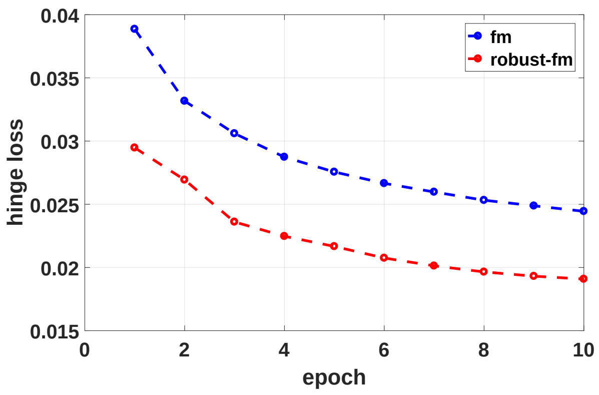
Experimental Results

phishing	Train loss	Test loss	Acc(%)
LR	±	±	±
SVM	0.1451±0.0002	0.1590±0.0058	93.26 ± 0.30
FM	0.1397±0.0002	0.1534±0.0006	93.35 ± 0.10
RobFM-IncSVD	±	±	±
RobFM	**0.1145±0.0004**	**0.1357±0.0051**	**94.59 ± 0.33**
w8a	Train loss	Test loss	Acc(%)
LR	±	±	±
SVM	0.0305±0.0006	0.0316±0.0003	98.65 ± 0.02
FM	0.0234±0.0010	0.0245±0.0002	98.86 ± 0.07
RobFM-IncSVD	±	±	±
RobFM	**0.0178±0.0007**	**0.0190±0.0003**	**99.10 ± 0.06**
protein	Train loss	Test loss	Acc(%)
LR	±	±	±
SVM	1.5369±0.0002	1.5198±0.0014	68.61 ± 0.07
FM	1.3757±0.0002	1.4668±0.0016	69.21 ± 0.10
RobFM-IncSVD	±	±	±
RobFM	±	±	±
IJCNN	Train loss	Test loss	Acc(%)
LR	±	±	±
SVM	0.1770±0.0004	0.1843±0.0003	91.35 ± 0.02
FM	0.0930±0.0005	0.0955±0.0004	96.66 ± 0.09
RobFM-IncSVD	±	±	±
RobFM	**0.0712±0.0001**	**0.0744±0.0013**	**97.83 ± 0.15**
Covtype	Train loss	Test loss	Acc(%)
LR	±	±	±
SVM	0.6080±0.0003	0.6102±0.0002	68.76 ± 0.06
FM	0.5111±0.0001	0.5088±0.0030	77.31 ± 0.10
RobFM-IncSVD	±	±	±
RobFM	**0.4894±0.0001**	**0.4844±0.0066**	**79.65 ± 0.25**
MNIST	Train loss	Test loss	Acc(%)
LR	±	±	±
SVM	0.5957±0.0008	0.5776±0.0007	91.64 ± 0.13
FM	0.2224±0.0005	0.2949±0.0034	96.62 ± 0.08
RobFM-IncSVD	±	±	±
RobFM	±	±	±

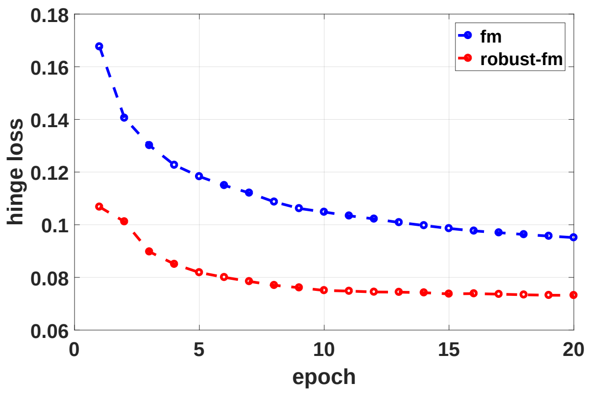
Table 2: Comparison of different algorithms in terms of train loss, test loss, classification accuracy



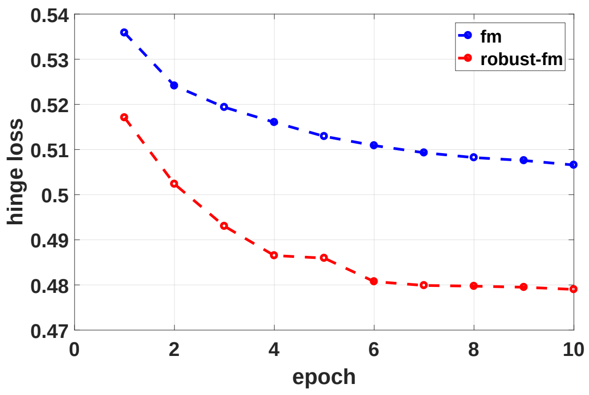
(a) phishing



(b) w8a



(c) IJCNN



(d) Covtype

Figure 1: Epoch-wise demonstration of different algorithms with hinge loss on test data