Robust Factorization Machines for User Response Prediction

Surabhi Punjabi*
@WalmartLabs, Bangalore, India
surabhi.punjabi@gmail.com

Priyanka Bhatt*
@WalmartLabs, Bangalore, India
priyankabhatt91@gmail.com

ABSTRACT

Factorization machines (FMs) are a state-of-the-art model class for user response prediction in the computational advertising domain. Rapid growth of internet and mobile device usage has given rise to multiple customer touchpoints. This coupled with factors like high cookie churn rate results in a fragmented view of user activity at the advertiser's end. Current literature assumes procured user signals as the absolute truth, which is contested by the absence of deterministic identity linkage across a user's multiple avatars. In this work, we characterize the data uncertainty using Robust Optimization (RO) paradigm to design approaches that are immune against perturbations. We propose two novel algorithms: robust factorization machine (RFM) and its field-aware variant (RFFM), under interval uncertainty. These formulations are generic and can find applicability in any classification setting under noise. We provide a distributed and scalable Spark implementation using parallel stochastic gradient descent. In the experiments conducted on three real-world datasets, the robust counterparts outperform the baselines significantly under perturbed settings. Our experimental findings reveal interesting connections between choice of uncertainty set and the noise-proofness of resulting models.

KEYWORDS

Factorization Machines; Field-aware Factorization Machines; Robust Optimization; Computational Advertising; Response Prediction; Interval Uncertainty

ACM Reference Format:
Surabhi Punjabi and Priyanka Bhatt. 2018. Robust Factorization Machines for User Response Prediction. In *WWW 2018: The 2018 Web Conference, April 23–27, 2018, Lyon, France.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3178876.3186148

1 INTRODUCTION

User response prediction is a central problem in the computational advertising domain. The primary stakeholders in this ecosystem are: publishers who possess ad inventory, advertisers who bid for these ad slots and users who are exposed to the ad experience. Publishers and advertisers leverage signals like online user footprint, demographics and associated context for modeling user intent. Clicks and conversions being the key objectives, response prediction problem is generally formulated as estimating the probability of click or conversion given an ad impression. This probability subsequently translates to user-level bids or manifests itself in creating discrete user segments according to the propensity of user intent. This area has garnered interest from both the industry and academia, with logistic regression (LR) [4, 10] being the conventional choice. The recently proposed factorization machines (FMs) [22] and field-aware factorization machines (FFMs) [13] which model feature interactions in latent space have outperformed LR on several experimental and production datasets [1, 12].

User interaction signals, which serve as primary input to the predictive modeling, are procured from a wide variety of online sources like social media, search engines, e-commerce platforms, and news portals. A user's activity might be spread across multiple devices like desktop, mobile, and tablet. Interestingly, users exhibit a manifold of browsing patterns and device specific avatars: a user seeming to be a keen shopper on desktop might just be a casual browser on mobile. Without cross device linkages in place, the user interaction dataset comprises of multiple incomplete views for the same user. For US, Criteo [6] estimates that a whopping 31% of online transactions involve two or more devices and that both the conversion rates and buyer-journeys increase by about 40% in user-centric view of activity across multiple devices as compared to a partial device-specific view.

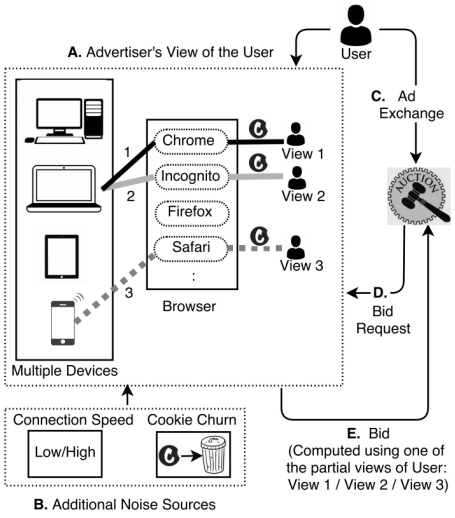


Figure 1: (A) User visits an advertiser's site several times on different devices and/or browsers: the advertiser observes multiple fragmented views of user's real activity. (B) Additional noise induced during data collection. (C, D) User visiting a publisher's site results in an online auction and a bid request is sent to different advertisers. (E) Advertiser responds with a bid computed using data from only one of the user views.

Even for the same device, factors like operating system, network connectivity and browser type have their own associated data leakages. These heterogeneities in the underlying generative mechanism lead to noise in the collected data. However, this problem has

*Both authors contributed equally to the paper

WWW 2018, April 23–27, 2018, Lyon, France

ACM ISBN 978-1-4503-5639-8/18/04.
https://doi.org/10.1145/3178876.3186148

been heavily overlooked with only 5% marketers having seamlessly integrated customer touchpoints [8]. With around 65% cookies being deleted monthly [5], cookie churn acts as an additional contributor to noise and makes the user-advertiser association short lived. See Figure 1.

These cumulative inefficiencies during data collection end up camouflaging the user's holistic identity and raise a compelling question on the quality of the same data hose that had generated these signals in the first place. Some recent works have attempted to probabilistically stitch the user identity [14, 23], but a complete consolidation of user profiles remains an open problem.

Objective functions in the existing response prediction literature assume user profiles to be precisely known and remain agnostic to the inherent noise present in the input signals [25]. Consequently, the learnt classifiers possess a fuzzy understanding of the underlying causal relationships and thus exhibit remarkable sensitivity towards small data perturbations. Since model predictions guide bid price determination, a monetary loss or an opportunity cost is incurred for every misclassification [15].

This work aims at characterizing the environment-induced uncertainty in the user signals and reformulating the FM and FFM objective functions to be immune against data fluctuations. For this we utilize the robust optimization (RO) framework [2] which assumes a deterministic, set based uncertainty model and seeks solutions that are computationally tractable and remain near-optimal in the worst case realization of uncertainty. To the best of our knowledge, this is the first work advocating the application of RO in the user modeling domain. The main contributions of this paper are summarized below:

- We employ robust optimization principles to model the noise arising in online advertising signals as bounded box-type interval uncertainty sets.
- We propose two novel algorithms: robust factorization machine (RFM) and robust field-aware factorization machine (RFFM), as robust minimax formulations for FM and FFM respectively.
- We provide a distributed and scalable Spark based implementation for solving the optimization problem using parallel stochastic gradient decent.
- We present a comprehensive evaluation of our formulations on three publicly available response prediction datasets from Criteo and Avazu. The price of robustness is a classifier which takes a slight performance hit in the standard setting (-0.24% to -1.1%) but significantly outperforms the non-robust counterparts (4.45% to 38.65%) when subjected to noise.
- We systematically assess the tradeoff between robustness under noise and optimality of performance in standard setting and provide guidelines for selection of uncertainty sets.
- We extensively study model calibration and the effects of hyperparameters, initialization strategies, and parallelism on model performance and convergence.
- The final formulations obtained are generic and can aid in any noise sensitive classification domain. To demonstrate this broad applicability, we present results on a credit card fraud detection dataset.

2 PRELIMINARIES

2.1 Response Prediction

We begin with an overview of the state-of-the-art approaches for predicting user propensity to click or convert given ad exposure. This is a supervised learning setting wherein the learner is presented with a set of m training instances $\{(\mathbf{x}^{(i)}, y^{(i)}) | \mathbf{x}^{(i)} \in \mathbb{R}^d, y^{(i)} \in \{1, -1\}\ \forall i \in \{1, \ldots, m\}\}$, where $\mathbf{x}^{(i)}$ represents activity and context signals for user i and $y^{(i)}$ is the binary response variable which captures whether or not the user subsequently clicked or converted.

Logistic regression has long been the preferred classifier for user response modeling [4, 10] since it offers the advantage of well calibrated probability outputs, is highly scalable and also yields interpretable models. It learns weight vector $\mathbf{w} \in \mathbb{R}^d$ by maximizing log likelihood against a regularization penalty. The corresponding loss minimization equivalent takes the following form:

$$\min_{\mathbf{w}} \frac{\lambda}{2}\|\mathbf{w}\|_2^2 + \frac{1}{m}\sum_{i=1}^{m} \log\left(1 + \exp(-y^{(i)}\phi(\mathbf{x}^{(i)}, \mathbf{w}))\right) \quad (1)$$

$$\text{where, } \phi(\mathbf{x}, \mathbf{w}) = w_0 + \sum_{j=1}^{d} w_j x_j.$$

Linear models suffer with the limitation of not being able to model the effect of feature interactions on the dependent variable. Factorization machines (FMs) proposed in [22] have recently gained popularity as an effective learning paradigm for capturing impact of feature conjunctions especially for sparse datasets. They aim at learning a projection $\mathbf{v}_j$ for every feature j in a latent factor space $\mathbb{R}^p$. The strength of feature interactions is quantified by the inner product of the corresponding factors. The optimization problem remains similar to (1), with ϕ evolving to ϕ_{FM} to incorporate these interaction terms:

$$\phi_{FM}(\mathbf{x}, \mathbf{w}, \mathbf{V}) = w_0 + \sum_{j=1}^{d} w_j x_j + \sum_{j=1}^{d}\sum_{k=j+1}^{d} \langle \mathbf{v}_j, \mathbf{v}_k \rangle x_j x_k \quad (2)$$

where $\langle . \rangle$ represents inner product and $\mathbf{V} \in \mathbb{R}^{d \times p}$ is a factor matrix composed of the latent vectors $\mathbf{v}_j$. Response prediction datasets largely comprise of categorical features a.k.a. fields. Typical examples of fields are publisher, device, brand, etc. which may take values from sets {CNN, Vogue}, {desktop, mobile, tablet}, and {Nike, Adidas} respectively. LR and FM use the expanded feature space generated by one-hot encoding of these categorical variables and the semantics of the 'field' itself are lost.

Field-aware factorization machines (FFMs) are a model class that leverage the field information associated with every feature and extend the concept of FM to learn a dedicated latent vector corresponding to every (feature, field) combination [12]. So instead of learning a latent vector per feature, i.e. $\mathrm{v}_{\mathrm{Vogue}}$, $\mathrm{v}_{\mathrm{Nike}}$, etc. the model learns separate latent vectors for capturing fieldwise interactions like $\mathrm{v}_{(\mathrm{Vogue,\ device})}$, $\mathrm{v}_{(\mathrm{Vogue,\ brand})}$, etc. The function ϕ thus further evolves to:

$$\phi_{FFM}(\mathbf{x}, \mathbf{w}, \mathbf{V}) = w_0 + \sum_{j=1}^{d} w_j x_j + \sum_{j=1}^{d}\sum_{k=j+1}^{d} \langle \mathbf{v}_{j,f_k}, \mathbf{v}_{k,f_j} \rangle x_j x_k \quad (3)$$

where $\mathbf{v}_{j,f_k} \in \mathbb{R}^p$ is a latent vector capturing interactions between feature j and field of feature k and $\mathbf{V} \in \mathbb{R}^{d \times q \times p}$ is a tensor composed of all these $\mathbf{v}_{j,f_k}$ vectors. Here q denotes the number of fields in the dataset.

FMs and FFMs have demonstrated superior generalization ability against other classifiers by winning two Kaggle competitions[1,2] in the past. Other techniques like model ensembles [9, 27] and deep learning [3, 21] have also been explored for the user response prediction task. For this work, we restrict our focus on formulating robust counterparts for FM and FFM models.

2.2 Robust Optimization

Traditional stochastic optimization provides probabilistic characterization of noise in measurements. In contrast, paradigm of Robust Optimization (RO) [2] models uncertainty as bounded set based variability in the input observations. The uncertainty set is defined as $U = \{\boldsymbol{\mu}^{(i)} | \mathbf{x}^{(i)} - \boldsymbol{\eta}^{(i)} \leq \mathbf{x}^{(i)} + \boldsymbol{\mu}^{(i)} \leq \mathbf{x}^{(i)} + \boldsymbol{\eta}^{(i)}, \forall i \in \{1, \ldots, m\}\}$. Here $\boldsymbol{\eta}^{(i)} \in \mathbb{R}^d_{\geq 0}$ represents the uncertainty bound for input $\mathbf{x}^{(i)}$. Incorporating this notion of deterministic uncertainty allows for multiple manifestations of the input data points anywhere within the specified bounds. RO seeks to learn a function that remains feasible and near optimal for all possible realizations of uncertainty. For classification setting, this translates to minimizing the worst case loss suffered against all possible data perturbations. Assuming a general loss function $\mathcal{L}(\mathbf{w}, \mathbf{X})$, the robust counterpart takes the following minimax form:

$$\min_{\mathbf{w}} \max_{U} \frac{1}{m} \sum_{i=1}^{m} \mathcal{L}(\mathbf{w}, \mathbf{x}^{(i)} + \boldsymbol{\mu}^{(i)}). \tag{4}$$

Computational tractability of robust formulations is governed by the choice of uncertainty sets. Box-type, ellipsoidal, conic, and polyhedral are commonly employed classes of uncertainty sets in the RO literature [11]. In this work, we design robust formulations assuming box-type (or interval) uncertainty under which, for each observation $\mathbf{x} \in \mathbb{R}^d$, there is a corresponding uncertainty vector $\boldsymbol{\mu} \in \mathbb{R}^d$ such that each dimension of the vector is bounded independently, i.e. $|\mu_j| \leq \eta_j, \forall j \in \{1, \ldots, d\}$. The choice of interval uncertainty facilitates noise independence amongst individual features. Geometrically, this can be visualized as data points residing in a bounded hyperrectangular manifold.

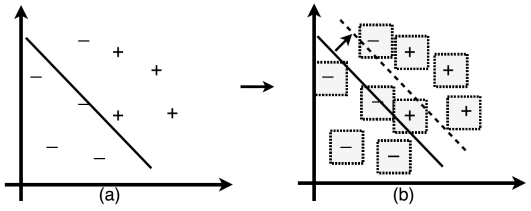


Figure 2: (a) Classifier trained on original data. (b) When box-type uncertainty is associated with data points, the learnt classifier boundary shifts in order to accommodate the effect of perturbations.

Figure 2 illustrates how the training instances appear to a learner in a standard setting and after introducing box-type uncertainty. Note the shift in decision boundary of the learnt classifier. The RO framework presents a systematic tradeoff between choosing optimal classifier weights for given observations and robustifying against perturbations. Robust formulations for LR and support vector machines (SVMs) have been proposed in [7, 16]. Our work is the first attempt to systematically introduce robustness in the factorization machines.

[1]https://www.kaggle.com/c/criteo-display-ad-challenge
[2]https://www.kaggle.com/c/avazu-ctr-prediction

3 PROPOSED APPROACH

FM and its extensions have witnessed a rapid adoption recently, not just within the purview of Kaggle competitions but also for the real-world bidding systems [12]. To incorporate noise-proofness against data perturbations in these models, we design robust counterparts for FM and FFM using RO principles under interval uncertainty. The resulting minimax formulation is then reduced to a pure minimization problem by obtaining upper bounds on terms involving uncertainty. We propose a stochastic gradient descent based parallelized training algorithm which can be deployed on a distributed environment like Spark for learning the final weight matrices.

3.1 Robust FM

Factorization machines consider both linear and pairwise feature interactions. This presents us with a choice of either sharing the same uncertainty vectors across the two types of interactions or decoupling the uncertainty parameters. We go with the second alternative and for each data point $\mathbf{x}$, we associate uncertainty vector $\boldsymbol{\mu} \in \mathbb{R}^d$ s.t. $|\mu_j| \leq \eta_j, \forall j \in \{1, \ldots, d\}$ for characterizing noise in linear interactions and matrix $\Sigma \in \mathbb{R}^{d \times d}$ s.t. $\Sigma_{j,k} = \sigma_j \sigma_k$, $|\sigma_j| \leq \rho_j, \forall j \in \{1, \ldots, d\}$ for capturing noise induced by pairwise interaction terms. This choice is motivated by two reasons. The presence of Σ offers another degree of freedom while tuning the model. Also, order-2 interactions are being learnt in a latent space, which might not have similar semantics as the original feature space. This definition of $\boldsymbol{\mu}$ and Σ confines the hyperparameter space to be linear in the number of features for a given training example. We now introduce these uncertainty terms and formulate ϕ for robust factorization machines (RFMs) as ϕ_{RFM}. For mathematical convenience, we add self-interaction terms to the robust variant.

$$\begin{aligned}
&\phi_{RFM}(\mathbf{x}, \mathbf{w}, \mathbf{V}, \boldsymbol{\mu}, \Sigma) \\
=&w_0 + \sum_{j=1}^{d} w_j(x_j + \mu_j) + \sum_{j=1}^{d} \sum_{k=j}^{d} \langle \mathbf{v}_j, \mathbf{v}_k \rangle (x_j x_k + \Sigma_{j,k}) \\
=&w_0 + \sum_{j=1}^{d} w_j(x_j + \mu_j) + \frac{1}{2} \sum_{j=1}^{d} \sum_{k=1}^{d} \langle \mathbf{v}_j, \mathbf{v}_k \rangle (x_j x_k + \Sigma_{j,k}) \\
&+ \frac{1}{2} \sum_{j=1}^{d} \langle \mathbf{v}_j, \mathbf{v}_j \rangle (x_j^2 + \Sigma_{j,j}) \quad \text{(Rearranging terms)} \\
=&w_0 + \sum_{j=1}^{d} w_j x_j + \sum_{j=1}^{d} w_j \mu_j + \frac{1}{2} \sum_{j=1}^{d} \sum_{k=1}^{d} \langle \mathbf{v}_j, \mathbf{v}_k \rangle x_j x_k + \\
&\frac{1}{2} \sum_{j=1}^{d} \sum_{k=1}^{d} \langle \mathbf{v}_j, \mathbf{v}_k \rangle \Sigma_{j,k} + \frac{1}{2} \sum_{j=1}^{d} \langle \mathbf{v}_j, \mathbf{v}_j \rangle x_j^2 + \frac{1}{2} \sum_{j=1}^{d} \langle \mathbf{v}_j, \mathbf{v}_j \rangle \Sigma_{j,j} \\
=&w_0 + \sum_{j=1}^{d} w_j x_j + \sum_{j=1}^{d} w_j \mu_j + \frac{1}{2} \sum_{j=1}^{d} \sum_{k=1}^{d} \sum_{f=1}^{p} v_{j,f} v_{k,f} x_j x_k + \\
&\frac{1}{2} \sum_{j=1}^{d} \sum_{k=1}^{d} \sum_{f=1}^{p} v_{j,f} v_{k,f} \Sigma_{j,k} + \frac{1}{2} \sum_{j=1}^{d} \sum_{f=1}^{p} v_{j,f}^2 x_j^2 + \frac{1}{2} \sum_{j=1}^{d} \sum_{f=1}^{p} v_{j,f}^2 \Sigma_{j,j} \\
&\text{(Expanding terms along the factor space)} \\
=&w_0 + \sum_{j=1}^{d} w_j x_j + \sum_{j=1}^{d} w_j \mu_j + \frac{1}{2} \sum_{f=1}^{p} (\sum_{j=1}^{d} v_{j,f} x_j)^2 + \\
&\frac{1}{2} \sum_{f=1}^{p} (\sum_{j=1}^{d} v_{j,f} \sigma_j)^2 + \frac{1}{2} \sum_{f=1}^{p} \sum_{j=1}^{d} v_{j,f}^2 x_j^2 + \frac{1}{2} \sum_{f=1}^{p} \sum_{j=1}^{d} v_{j,f}^2 \sigma_j^2
\end{aligned}$$

With the redefined ϕ for RFM, the loss minimization view under uncertainties results in the following minimax formulation:

$$\min_{\mathbf{w},\mathbf{V}} \max_{\substack{\boldsymbol{\mu}^{(j)},\Sigma^{(j)} \\ \forall 1\le j\le m}} \frac{\lambda}{2}\|\mathbf{w}\|_2^2 + \frac{\lambda}{2}\|\mathbf{V}\|_2^2 + \frac{1}{m}\sum_{i=1}^{m}\log\left(1+\exp(\Omega^i_{RFM})\right) \quad (5)$$

where, Ω^i_{RFM} is shorthand for $\Omega_{RFM}(\mathbf{x}^{(i)}, y^{(i)}, \mathbf{w}, \mathbf{V}, \boldsymbol{\mu}^{(i)}, \Sigma^{(i)})$ and, $\Omega_{RFM}(\mathbf{x}, y, \mathbf{w}, \mathbf{V}, \boldsymbol{\mu}, \Sigma) = -y\,\phi_{RFM}(\mathbf{x}, \mathbf{w}, \mathbf{V}, \boldsymbol{\mu}, \Sigma)$.

The inner maximization signifies the worst case loss incurred because of the uncertainty parameters $\boldsymbol{\mu}^{(j)}$ and $\Sigma^{(j)}$, $\forall\ 1 \le j \le m$. Due to monotonicity of the terms in summation, maximizing the objective function in (5) reduces to maximizing Ω_{RFM}. We hereby refer to the optimal solution of the reduced subproblem as Ω^{wc}_{RFM}.

$$\Omega^{wc}_{RFM}(\mathbf{x}, y, \mathbf{w}, \mathbf{V}) = \max_{\boldsymbol{\mu},\Sigma} \Omega_{RFM}(\mathbf{x}, y, \mathbf{w}, \mathbf{V}, \boldsymbol{\mu}, \Sigma) \quad (6)$$

Further, we derive the value of Ω^{wc}_{RFM} by obtaining upper bounds on the terms with uncertainties. Since the linear and pairwise uncertainty parameters have been considered independent, we can examine the relevant terms in isolation. We first group and analyze the terms associated with pairwise uncertainty σ.

$$\begin{aligned} &-y\frac{1}{2}\Big(\sum_{f=1}^{p}\Big(\sum_{j=1}^{d} v_{j,f}\sigma_j\Big)^2 + \sum_{f=1}^{p}\sum_{j=1}^{d} v_{j,f}^2\sigma_j^2\Big) \\ &\le \frac{1}{2}\Big(\sum_{f=1}^{p}\Big(\sum_{j=1}^{d} |v_{j,f}||\sigma_j|\Big)^2 + \sum_{f=1}^{p}\sum_{j=1}^{d} v_{j,f}^2\sigma_j^2\Big) \\ &\le \frac{1}{2}\Big(\sum_{f=1}^{p}\Big(\sum_{j=1}^{d} |v_{j,f}|\rho_j\Big)^2 + \sum_{f=1}^{p}\sum_{j=1}^{d} v_{j,f}^2\rho_j^2\Big) \end{aligned} \quad (7)$$

The last inequality follows from the definition of interval uncertainty, where all covariates are bounded independently. Similarly for the linear uncertainty terms we have,

$$-y\sum_{j=1}^{d} w_j\mu_j \le \sum_{j=1}^{d}|w_j||\mu_j| \le \sum_{j=1}^{d}|w_j|\eta_j. \quad (8)$$

Using the upper bounds obtained from (7) and (8) on uncertainty terms, we derive the value for Ω^{wc}_{RFM} as:

$$\begin{aligned} \Omega^{wc}_{RFM}(\mathbf{x}, y, \mathbf{w}, \mathbf{V}) = -yw_0 + \sum_{j=1}^{d}(-yw_jx_j + |w_j|\eta_j) - \frac{y}{2}\sum_{f=1}^{p}\Big(\sum_{j=1}^{d} v_{j,f}x_j\Big)^2 \\ + \frac{1}{2}\sum_{f=1}^{p}\Big(\sum_{j=1}^{d}|v_{j,f}|\rho_j\Big)^2 - \frac{y}{2}\sum_{f=1}^{p}\sum_{j=1}^{d} v_{j,f}^2x_j^2 + \frac{1}{2}\sum_{f=1}^{p}\sum_{j=1}^{d} v_{j,f}^2\rho_j^2. \end{aligned}$$

For notational convenience, we hereby refer to $\Omega^{wc}_{RFM}(\mathbf{x}, y, \mathbf{w}, \mathbf{V})$ as $\Omega^{wc}_{RFM}(\mathbf{x}, y)$. Using the derived value for Ω^{wc}_{RFM} the optimization problem in (5) simplifies to:

$$\min_{\mathbf{w},V} \frac{\lambda}{2}\|\mathbf{w}\|_2^2 + \frac{\lambda}{2}\|\mathbf{V}\|_2^2 + \frac{1}{m}\sum_{i=1}^{m}\log\left(1+\exp(\Omega^{wc}_{RFM}(\mathbf{x}^{(i)}, y^{(i)}))\right). \quad (9)$$

Note that by minimizing the worst case loss, we are encoding pessimism in the classifier, the magnitude of which varies with the size of hyperrectangles in which the data is bounded. Table 1 summarizes the notations used in this paper.

Table 1: Table of Notations

Notation	Description
$\mathbf{x}^{(i)}$ (or $\mathbf{x}$)	Feature vector $\in \mathbb{R}^d$ for sample i
$y^{(i)}$ (or y)	Label $\in \{-1, 1\}$ for sample i
x_j	Feature value $\in \mathbb{R}$ for j^{th} dimension in a sample
t	Number of epochs
m	Number of training samples
q	Number of fields in the feature data
d	Dimensionality of original feature space
p	Dimensionality of latent factors learnt per feature
α	Learning rate for stochastic gradient descent
λ	Regularization parameter
$\mathbf{w}$	Weight vector $\in \mathbb{R}^d$ for linear interactions
$\mathbf{V}$	Factor matrix $\in \mathbb{R}^{d\times p}$ for FM and tensor $\in \mathbb{R}^{d\times q\times p}$ for FFM
$\boldsymbol{\mu}^{(i)}$ (or $\boldsymbol{\mu}$)	Linear uncertainty vector $\in \mathbb{R}^d$ for a sample
$\Sigma^{(i)}$ (or Σ)	Pairwise uncertainty matrix $\in \mathbb{R}^{d\times d}$ for a sample
$\boldsymbol{\eta}^{(i)}$ (or $\boldsymbol{\eta}$)	Linear uncertainty bound $\in \mathbb{R}^d_{\ge 0}$ for a sample
$\boldsymbol{\rho}^{(i)}$ (or $\boldsymbol{\rho}$)	Simplified pairwise uncertainty bound $\in \mathbb{R}^d_{\ge 0}$ for a sample

3.2 Parameter Learning: Robust FM

We use minibatch stochastic gradient descent (SGD) to solve the optimization problem (9) for robust FM. The corresponding loss gradient is given by:

$$\frac{\delta\mathcal{L}}{\delta\theta} = \lambda\theta + \frac{1}{m}\sum_{i=1}^{m}\frac{\exp(\Omega^{wc}_{RFM}(\mathbf{x}^{(i)}, y^{(i)}))}{1+\exp(\Omega^{wc}_{RFM}(\mathbf{x}^{(i)}, y^{(i)}))} * \frac{\delta}{\delta\theta}(\Omega^{wc}_{RFM}(\mathbf{x}^{(i)}, y^{(i)})) \quad (10)$$

where,

$$\frac{\delta}{\delta\theta}(\Omega^{wc}_{RFM}(\mathbf{x}, y)) = \begin{cases} -y & \text{if } \theta = w_0 \\ -yx_j + \eta_j\text{sgn}(w_j) & \text{if } \theta = w_j \\ -yx_j\sum_{k=1}^{d} v_{k,f}x_k - yv_{j,f}x_j^2 + & \\ \rho_j\text{sgn}(v_{j,f})\sum_{k=1}^{d}|v_{k,f}|\rho_k + v_{j,f}\rho_j^2 & \text{if } \theta = v_{j,f}. \end{cases}$$

Here sgn(.) represents the sign function. Note that the update rule is composed of deterministic and uncertainty terms, the latter being independent of y since we have arrived at the formulation by maximizing Ω. The terms $\sum_{k=1}^{d} v_{k,f}x_k$ and $\sum_{k=1}^{d}|v_{k,f}|\rho_k$ being independent of j can be computed in advance. The details of our approach are outlined in Algorithm 1.

3.3 Robust FFM

We now derive the robust counterpart for field-aware factorization machine (FFM), the more rigorous and expressive variant of FMs. Incorporating linear and pairwise uncertainty parameters in the original function ϕ_{FFM} in equation (3) yields ϕ_{RFFM}.

$$\begin{aligned} \phi_{RFFM}(\mathbf{x}, \mathbf{w}, \mathbf{V}, \boldsymbol{\mu}, \Sigma) = w_0 + \sum_{j=1}^{d} w_j(x_j + \mu_j) \\ + \sum_{j=1}^{d}\sum_{k=j}^{d}\langle \mathbf{v}_{j,f_k}, \mathbf{v}_{k,f_j}\rangle(x_jx_k + \sigma_j\sigma_k) \end{aligned}$$

Note that $\mathbf{V}$ is a tensor comprising of latent vectors learnt per (feature, field) combination. Following steps as in Section 3.1 we

Algorithm 1: Robust FM

Input : **Training data** as a set of (Feature, Label) tuples $\{(\mathbf{x}, y)|\mathbf{x} \in \mathbb{R}^{\boldsymbol{d}}, y \in \{1,-1\}\}$
Uncertainty parameters: Linear and pairwise uncertainty bounds $(\boldsymbol{\eta}, \boldsymbol{\rho})$ for each data point
Hyperparameters: [$\boldsymbol{t}$ (#epochs), $\boldsymbol{p}$ (dimension of latent vectors), $\boldsymbol{r}$ (#data partitions) , $\boldsymbol{\alpha}$ (learning rate), $\boldsymbol{f}$ (initialization technique)]
Output : $\mathbf{w}, \mathbf{V}$

Initialize $\mathbf{w}, \mathbf{V}$ using f
Randomly split data over r partitions
for *epoch* $\in \{1, \ldots, t\}$ **do**
  for all *data partitions* $\in \{1, \ldots, r\}$ **parallel do**
    for *each sampled data point* $(\mathbf{x}, y)$ *in partition* **do**
      $w_0 := w_0 - \alpha * \frac{\delta \mathcal{L}}{\delta w_0}$
      for j *in* $\{1, \ldots, d\}$ **do**
        $w_j := w_j - \alpha * \frac{\delta \mathcal{L}}{\delta w_j}$
        for f *in* $\{1, \ldots, p\}$ **do**
          $v_{j,f} := v_{j,f} - \alpha * \frac{\delta \mathcal{L}}{\delta v_{j,f}}$
        end
      end
    end
  end
  $\mathbf{w} \leftarrow$ Average($\mathbf{w}$) over all r partitions
  $\mathbf{V} \leftarrow$ Average($\mathbf{V}$) over all r partitions
end

derive an upper bound on $-y\phi_{RFFM}$ (or Ω_{RFFM}), given by Ω^{wc}_{RFFM}.

$$\Omega^{wc}_{RFFM}(\mathbf{x}, y, \mathbf{w}, \mathbf{V}) = -yw_0 + \sum_{j=1}^{d}(-yw_jx_j + |w_j|\eta_j) - y\sum_{j=1}^{d}\sum_{k=j}^{d}\langle \mathbf{v}_{j,f_k}, \mathbf{v}_{k,f_j}\rangle x_jx_k + \sum_{j=1}^{d}\sum_{k=j}^{d}\langle |\mathbf{v}_{j,f_k}|, |\mathbf{v}_{k,f_j}|\rangle \rho_j\rho_k$$

Replacing Ω^{wc}_{RFM} with Ω^{wc}_{RFFM} in (9) gives the loss minimization problem for RFFM.

3.4 Parameter Learning: Robust FFM

We employ stochastic gradient descent (SGD) for parameter estimation. The gradient of loss function is given by:

$$\frac{\delta \mathcal{L}}{\delta \theta} = \lambda\theta + \frac{1}{m}\sum_{i=1}^{m}\frac{\exp(\Omega^{wc}_{RFFM}(\mathbf{x}^{(i)}, y^{(i)}))}{1+\exp(\Omega^{wc}_{RFFM}(\mathbf{x}^{(i)}, y^{(i)}))} * \frac{\delta}{\delta\theta}(\Omega^{wc}_{RFFM}(\mathbf{x}^{(i)}, y^{(i)})) \quad (11)$$

where,

$$\frac{\delta}{\delta\theta}\Omega^{wc}_{RFFM}(\mathbf{x}, y) = \begin{cases} -y & \text{if } \theta = w_0 \\ -yx_j + \eta_j \text{sgn}(w_j) & \text{if } \theta = w_j \\ -yv_{k,f_j}x_jx_k + & \\ \text{sgn}(v_{j,f_k})|v_{k,f_j}|\rho_j\rho_k & \text{if } \theta = v_{j,f_k}. \end{cases}$$

The final algorithm for RFFM differs from RFM in the core weight update steps in Algorithm 1. Also, similar to [13], we perform updates only on the non zero dimensions of the weight matrix to avoid unnecessary computation.

4 EXPERIMENTS

In this section we investigate the effectiveness of RFM and RFFM against their non-robust counterparts. In particular, we (i) evaluate the prediction quality of robust classifiers on original and perturbed datasets, (ii) examine the noise resilience arising from the choice of uncertainty sets, (iii) empirically compare different initialization strategies for the weight matrix, (iv) assess the impact of hyper-parameters on model performance, (v) explore isotonic regression for calibration of classifiers, and (vi) study model convergence rate with increased parallelism. Our experimental findings reveal that by incorporating the notion of robustness, the resulting classifiers take a slight performance hit for the unperturbed datasets, but outperform the original formulations significantly when presented with noisy measurements.

4.1 Experimental Setup

4.1.1 Dataset Description. We evaluate our formulations on three publicly available real-world datasets. These encompass both clickthrough rate (CTR) and conversion rate (CVR) prediction settings, which are two central problems for large scale user response prediction.

- **Criteo CTR Prediction**
  Released for a Kaggle competition in 2014, this dataset has become an important benchmark for CTR estimation. The training data comprises of 45 million ad impressions served to users along with their online footprint in the form of 13 integer features and 26 hashed categorical features. Label indicates whether a user subsequently clicked on the ad or not. One-hot encoding of the categorical variables results in a feature space of size $\sim 10^6$.
- **Avazu CTR Prediction**
  This dataset was released as part of a Kaggle challenge by Avazu advertising platform. It contains ten days of click-through data on mobile devices. The feature set comprises of signals like hour of the day, banner position, site id, device model, etc.
- **Criteo Conversion Logs**
  This dataset consists of conversion feedback signals for a portion of Criteo's ad traffic. In each row of the dataset, features represent an ad served to a user and a conversion timestamp label indicates when the user converted. If there is no purchase by the user, the field is empty. It is used widely for standardization of CVR algorithms.

The dataset statistics are summarized in Table 2. For brevity, we sometimes refer to Criteo click and conversion datasets as CriClick and CriConv respectively. In addition to the performance evaluation on these computational advertising datasets, we include a case study on a credit card fraud detection dataset in Section 5 to highlight that RFM and RFFM can characterize noise across domains.

Table 2: Summary Statistics of Datasets

Dataset	#Instances	#Features	#Fields
Criteo CTR Prediction	45,840,617	10^6	39
Avazu CTR Prediction	40,428,967	10^6	33
Criteo Conversion Logs	15,898,883	10^4	17

4.1.2 Evaluation Metric. For maximizing the efficiency of an ad campaign, the class probabilities estimated by a classifier need to be well calibrated since they directly impact the subsequent bidding for auctions. Hence we use logloss as the benchmarking metric for assessing model quality. Logloss (also known as logistic loss or cross entropy loss) is defined for the binary classification setting as:

$$-\frac{1}{m}\sum_{i=1}^{m} y^{(i)}\log(p^{(i)}) + (1-y^{(i)})\log(1-p^{(i)})$$

where $p^{(i)}$ is the probability or the confidence estimate assigned by a classifier for sample i belonging to the positive class, and $y^{(i)} \in \{0, 1\}$ is the true label. Logloss metric possesses an information theoretic interpretation of being the cross entropy between the true and predicted class distributions. An ideal model will have zero logloss. Lower values of this metric imply less divergence from true labels and hence superior model performance.

4.2 Implementation Details

We have implemented RFM and RFFM on Apache Spark [26], a distributed computing framework facilitating efficient parallelization, which is crucial for timely processing of the current massive datasets. Spark provides fault tolerant data storage abstraction: RDD (Resilient Distributed Dataset), which is an immutable collection of data partitioned across cluster nodes. The data is stored in-memory, which is highly favorable for iterative workloads.

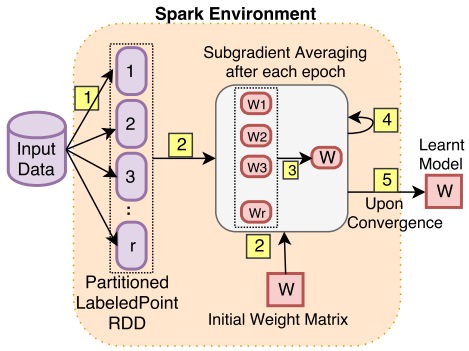


Figure 3: Spark implementation workflow for robust factorization machines.

We employ *iterative parameter mixing* strategy for performing distributed stochastic gradient descent (SGD) [17, 28]. Figure 3 outlines the implementation workflow. (1) Input data is partitioned across executors. (2) An initial weight matrix is broadcasted to all data partitions. Each node performs a single update of minibatch SGD on the subset of data it contains. (3) After every epoch, models learnt independently by the nodes are averaged. (4) The resulting global model is broadcasted again. (5) The algorithm terminates when error between successive iterations falls below a threshold. This distributed training strategy demonstrates fast convergence owing to the synchronous nature of model updates.

Memory requirement of our formulations is proportional to the number of features in a dataset. Distributed SGD adds an additional latency in terms of model transmission cost over the network after each epoch. Model compactness is therefore imperative for driving efficient performance. Owing to high dimensionality of the feature space, we resort to the hashing trick [4], which uses a simple hash function to restrict the number of feature values.

In our experimentation, we use 80% of the data for training and 10% each for constructing validation and test sets. Also, since our goal is to examine the difference between the robust and non-robust variants, we refrain ourselves from delving into feature engineering and focus exclusively on the model specific aspects. The code and dataset links for our implementation are available on Dropbox[3] for experimental reproducibility.

4.3 Choice of Uncertainty Sets

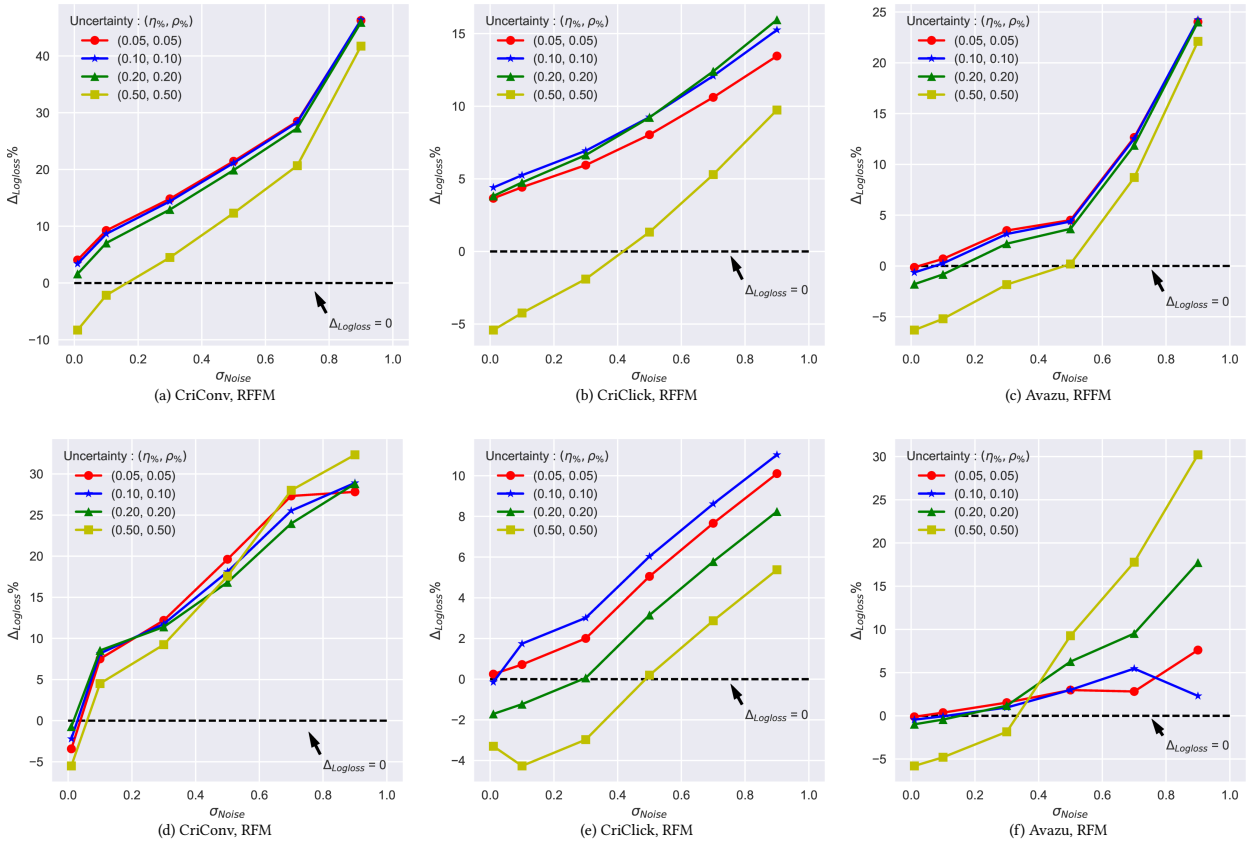
Having made the design choice of considering box-type uncertainty around data points in order to facilitate independence amongst covariates, the next critical step is to associate uncertainty bounds $(\boldsymbol{\eta}, \boldsymbol{\rho})$ with each training example. A straightforward approach is *absolute assignment* i.e. keeping these variables as additional parameters whose optimal values can be determined by parameter tuning. However, this is an expensive solution which would explode the hyperparameter space and render the possibility of obtaining the best performing model infeasible under practical settings. Another approach which seems appealing at first is to have *field specific* uncertainty values so that the number of newly introduced parameters is bounded. This approach however has the following drawbacks: it is tightly coupled with the dataset at hand and no direct link can be established between parameters selected and noise-proofness of the model procured after training.

These concerns encourage us to adopt the strategy of *relative assignment* for our experimentation. In this approach we select two positive real valued parameters $(\eta_{\%}, \rho_{\%})$ such that for every measurement γ, the effective linear and pairwise uncertainty bounds are given by $(\eta_{\%} * \gamma, \rho_{\%} * \gamma)$. This simple trick significantly brings down the size of parameters to tune and at the same time retains the feasibility of assigning variable sized hyperrectangles around the data. Under this formulation, larger measurements are associated with higher variability or lower confidence. Additionally, we threshold the uncertainty bounds to moderate the incorporated noise. As we shall present in the results below, this methodology of designing uncertainty sets has a nice interpretability in terms of cost incurred by incorporating robustness v/s resistance gained under noisy settings.

4.4 Performance Comparison

We compare the performance of RFM and RFFM models against the original factorization machine formulations on the conversion and click datasets. We particularly focus on the relative behavior under noisy settings. Gaussian distribution is a popular noise model in signal processing [24]. On similar lines, we simulate noise in original datasets by adding a Gaussian perturbation $\mathcal{N} \sim (\mu_{noise}, \sigma_{noise})$ to the test data. We vary the noise parameters and examine the goodness of classifiers for both original and perturbed versions of datasets. The results are presented in Table 3. By adhering to worst case loss minimization, the robust classifiers take a conservative view even for the original datasets, resulting in a higher logloss as compared to the non-robust equivalents. However, when subjected

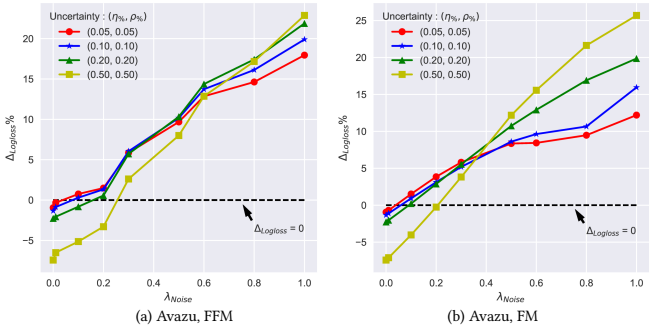
[3] https://www.dropbox.com/sh/ny6puvtopl98339/AACExLZ0waDL_ibWhfNItJfGa?dl=0



(a) CriConv, RFFM (b) CriClick, RFFM (c) Avazu, RFFM

(d) CriConv, RFM (e) CriClick, RFM (f) Avazu, RFM

Figure 4: Study of classifier behavior for different ($\eta_{\%}$, $\rho_{\%}$) when subjected to perturbed variants of the original dataset during the test phase. A Gaussian noise with μ_{noise} = 0.1 with varying $\sigma_{noise} \in \{0.01, 0.1, 0.3, 0.5, 0.7, 0.9\}$ is added to the test samples. Higher values of $\Delta_{logloss}$% indicates superior noise resilience.



(a) Avazu, FFM (b) Avazu, FM

Figure 5: Relative logloss reduction offered by robust formulations under Poisson noise for $\lambda_{noise} \in \{0.01, 0.1, 0.2, 0.3, 0.5, 0.6, 0.8, 1.0\}$.

to noise, the average performance degradation of robust classifiers is remarkably lower than the FMs and FFMs. In the risk sensitive domain of user modeling where signals captured might not be representative of complete user intent, this graceful degradation is a desirable property.

The levers offered by robust formulations for regulating uncertainty bounds are ($\eta_{\%}$, $\rho_{\%}$). Higher values imply higher uncertainty accounted for by the trained classifiers and hence greater immunity against noise. We train multiple RFM and RFFM models by varying ($\eta_{\%}$, $\rho_{\%}$) and study the relative reduction in logloss against the non-robust variants for different noise configurations. This relative reduction is given by $\Delta_{logloss}\% = \frac{(\mathcal{L}_{Original} - \mathcal{L}_{Robust})}{\mathcal{L}_{Original}} * 100$, where $\mathcal{L}_{Original}$ and $\mathcal{L}_{Robust}$ indicate loss under original and robust formulations respectively. As is evident from Figure 4, for each ($\eta_{\%}$, $\rho_{\%}$) the robust classifier starts off with a higher logloss w.r.t. baseline ($\Delta_{logloss}\% < 0$) when the σ_{noise} is low. However, on increasing the standard deviation of noise, the reduction in logloss is pronounced and goes as high as 40% for some cases. These findings not only reinforce the fact that the proposed classifiers indeed demonstrate superiority under noisy measurements, but also present an interesting tradeoff between aiming for high robustness and the price to be paid in the unperturbed settings.

In traditional signal processing systems, Poisson process is another widely used model for capturing noise. In an attempt to provide a comprehensive treatment to performance study under noise, we experiment with this noise model as well. Our findings are aligned with the insights procured for Gaussian noise model. We provide an interesting subset of results on the Avazu dataset in Figure 5.

These results reinforce the fact that the robust formulations are indeed able to withstand the potential incompleteness and corruption in response prediction datasets. Here we would like to reiterate the fact that our formulations are generic and can be applied to any domain where data uncertainty is a concern. The model designer can select ($\eta_{\%}$, $\rho_{\%}$) parameters in accordance with the degree of uncertainty, for the problem at hand.

Table 3: Comparative Analysis of Robust Formulations. $\mathcal{L}_M$ indicates logloss of model M. For Gaussian perturbation, the loss is averaged over $\mu_{noise} \in \{0.0001, 0.001, 0.01, 0.1, 1.0\}$, $\sigma_{noise} \in \{0.01, 0.1, 0.3, 0.5, 0.7, 0.9\}$ and over $\lambda_{noise} \in \{0.01, 0.1, 0.2, 0.3, 0.5, 0.6, 0.8, 1.0\}$ for Poisson noise model. Lower logloss indicates better model. Robust counterparts take performance hit for original datasets but outperform the base classifiers under perturbed settings.

	Perturbation / Parameters	Unperturbed			Gaussian			Poisson		
Dataset	(p, t, α, λ)	$\mathcal{L}_{FM}$	$\mathcal{L}_{RFM}$	$\Delta_{logloss}\%$	$\mathcal{L}_{FM}$	$\mathcal{L}_{RFM}$	$\Delta_{logloss}\%$	$\mathcal{L}_{FM}$	$\mathcal{L}_{RFM}$	$\Delta_{logloss}\%$
CriClick	$(25, 150, 0.01, 10^{-3})$	0.4482	0.4532	-1.1212	0.5075	0.4849	4.4481	0.5254	0.4892	6.8951
CriConv	$(15, 100, 0.1, 10^{-4})$	0.3756	0.3780	-0.6474	0.4950	0.4009	19.0055	0.4927	0.4106	16.6560
Avazu	$(15, 100, 0.01, 10^{-4})$	0.3902	0.3942	-1.0118	0.4578	0.4207	8.1108	0.4770	0.4480	6.0754

(a) RFM v/s FM

	Perturbation / Parameters	Unperturbed			Gaussian			Poisson		
Dataset	(p, t, α, λ)	$\mathcal{L}_{FFM}$	$\mathcal{L}_{RFFM}$	$\Delta_{logloss}\%$	$\mathcal{L}_{FFM}$	$\mathcal{L}_{RFFM}$	$\Delta_{logloss}\%$	$\mathcal{L}_{FFM}$	$\mathcal{L}_{RFFM}$	$\Delta_{logloss}\%$
CriClick	$(10, 150, 0.01, 10^{-3})$	0.4580	0.4592	-0.2467	0.5414	0.4945	8.6677	0.5514	0.5010	9.1365
CriConv	$(5, 50, 0.01, 10^{-4})$	0.3746	0.3770	-0.6254	0.6438	0.3949	38.6541	0.6379	0.4118	35.4508
Avazu	$(10, 50, 0.01, 10^{-3})$	0.3915	0.3939	-0.6258	0.4465	0.4082	8.5699	0.4492	0.4148	7.6468

(b) RFFM v/s FFM

4.5 Discussion

4.5.1 Initialization Strategy. Non-convexity of the optimization problem for factorization machines makes the selection of initial weight matrix pivotal to the optimality of results obtained. The traditional FMs employ Gaussian distribution for initializing model weights. Laplace distribution has recently been proposed [19] as a superior initializer owing to the fact that it has a higher peak than Gaussian and consequently results in a better fit for sparse datasets. For FFMs, sampling from Uniform distribution is another commonly adopted initialization approach [13]. We investigate the impact of different initialization strategies. The results summarized in Table 4 indicate that Laplace distribution outperforms Gaussian and Uniform distributions in terms of logloss for both RFM and RFFM across the three datasets.

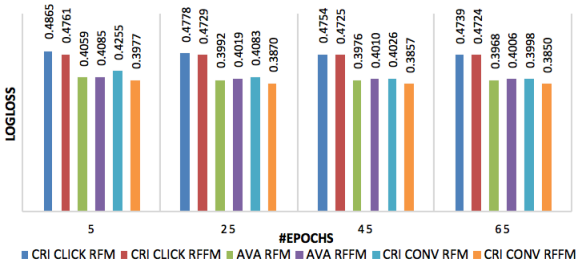
Table 4: Performance comparison of different initialization approaches. Logloss $\mathcal{L}_S$ is being incurred under initialization strategy S.

Dataset	Model	$\mathcal{L}_{Gaussian}$	$\mathcal{L}_{Laplace}$	$\mathcal{L}_{Uniform}$
CriClick	RFM	0.4538	0.4532	0.4541
	RFFM	0.4619	0.4582	0.4598
CriConv	RFM	0.3809	0.3780	0.3875
	RFFM	0.3784	0.3769	0.3774
Avazu	RFM	0.3944	0.3942	0.3952
	RFFM	0.3952	0.3939	0.3941

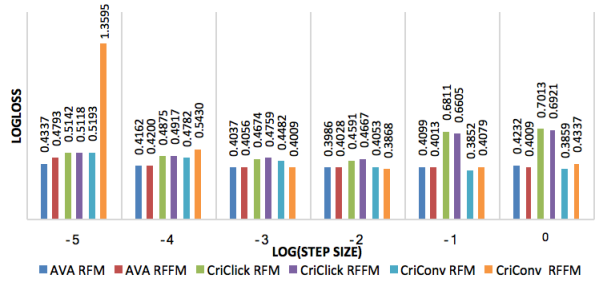
4.5.2 Impact of Hyperparameters. Parameter tuning is an important step for deriving optimal performance from any learning algorithm. From Figure 6(a), we observe that model (RFM/RFFM) performance improves with number of epochs t, though there is a diminishing returns property evident in improvement, which seems intuitive. This trend is consistent for all the datasets.

Gradient descent approaches are sensitive to the selection of learning rate α. As can be observed from Figure 6(b), for the same number of epochs, choosing smaller step size results in higher logloss since not enough exploration has been performed in the loss function landscape. On the other hand, selecting very high values of α might result in skipping the minima altogether leading to an increased loss value.

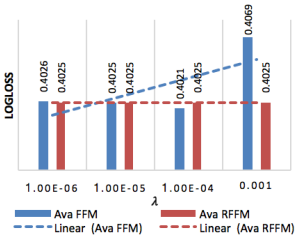
Regularization parameter plays a key role in preventing model overfitting. RFMs and RFFMs exhibit less sensitivity to changes in the value of λ as illustrated in Figure 6(c). This observation suggests that robustness inherently imposes some degree of regularization.



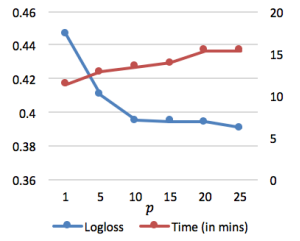
(a) Variation in logloss with Epochs



(b) Variation in logloss with Step size



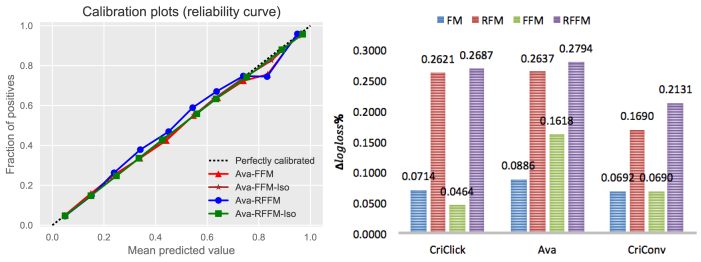
(c) Fluctuations in logloss with λ for Avazu dataset: FFM v/s RFFM



(d) Variation of training time and logloss with p for Avazu RFFM

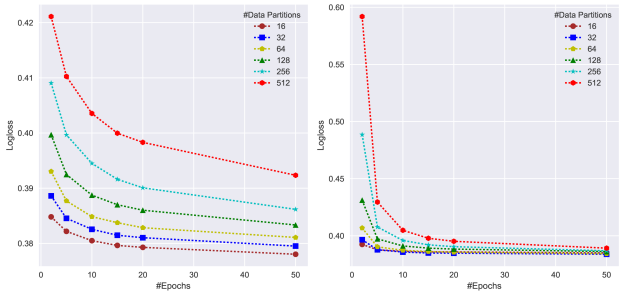
Figure 6: Effect of hyperparameters.

Higher number of latent factors p results in models possessing better generalization ability. However, for the distributed implementation of gradient descent, a large value of p translates into increased weight matrix serialization overhead and network communication cost for model synchronization among nodes. The reduction in logloss with p and the corresponding increase in training time are depicted in Figure 6(d).



(a) Calibration plot: Avazu FFM and RFFM (b) %Logloss reduction after calibration

Figure 7: Classifier calibration using isotonic regression.



(a) CriConv, RFM (b) CriConv, RFFM

Figure 8: Effect of parallelism on convergence rate.

4.5.3 Model Calibration. Training models that give accurate probabilistic outputs is central to the user modeling problem since it plays a major role in subsequent bidding. Calibration plots a.k.a. reliability diagrams serve as useful visualization tools for assessing goodness of model predictions with respect to true posterior probabilities. For each bucket of model prediction, the mean true probability is the fraction of positive samples in the bucket. The output of a perfectly calibrated classifier can be represented as a diagonal line on the reliability curve. To calibrate the model outputs, we employ isotonic regression technique [18] of univariate curve fitting with monotonicity constraints, where model outputs serve as regressors and actual label is the dependent variable.

We calibrate the probability estimates generated by RFMs, RFFMs and their non-robust counterparts and investigate the relative improvement in logloss. After this postprocessing, we observe a higher marginal improvement in the calibration quality of robust classifiers. Figure 7(a) depicts the reliability curve for Avazu FFM and RFFM before and after applying isotonic regression. The percentage logloss reduction achieved as a result of calibration for the three datasets is presented in Figure 7(b).

4.5.4 Impact of Parallelism on Model Convergence. Degree of parallelism has an inverse relationship with model convergence rate. Increasing the number of RDD partitions results in gradient descent being applied on smaller subsets of data and hence the averaged global model, procured after each epoch, is less stable. The downside of keeping lesser partitions is that each parallel worker is delegated with large number of samples, which increases the time taken per iteration. Figure 8 demonstrates the classic tradeoff between training time speedup v/s classification quality.

5 CASE STUDY: FRAUD DETECTION

The proposed formulations RFM and RFFM add a significant value for user response prediction under perturbed settings, as established by the experiments in Section 4. However, these are generic predictors, not restricted to the computational advertising domain and can be employed in any noise-sensitive classification scenario. To substantiate this claim, we test our formulations in the field of credit card fraud detection [20]. The dataset comprises of 284,807 anonymized credit card transactions and the challenge is to label them as fraudulent or genuine. The feature set is composed of 28 PCA transformed numerical variables with 0.172% of transactions labeled as fraud. Absence of categorical features renders RFM (or

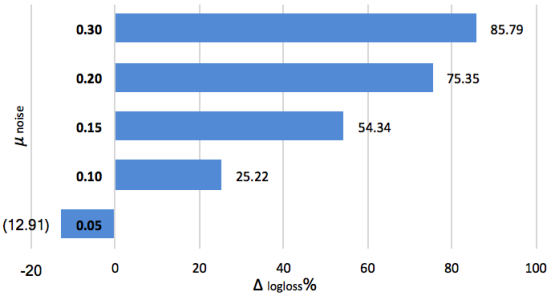


Figure 9: $\Delta_{logloss}$% for perturbed variants of fraud detection dataset where $\mu_{noise} \in \{0.05, 0.10, 0.15, 0.20, 0.30\}$. Logloss is averaged over $\sigma_{noise} \in \{0.001, 0.01, 0.1, 0.2, 0.3, 0.5, 0.7\}$ for each μ_{noise}.

FM) and RFFM (or FFM) formulations equivalent for this problem. The metric $\Delta_{logloss}$% (as defined in Section 4.4), which captures the logloss reduction achieved by the robust variants w.r.t. original FMs, exhibits dramatic improvement as the magnitude of Gaussian noise increases. This is evident from Figure 9, demonstrating superior noise resilience offered by RFMs. These findings are encouraging since they highlight domain independence of our formulations.

6 CONCLUSION AND FUTURE WORK

The ever increasing customer touchpoints and the associated noise sources have created a pressing need to design algorithms which take into account input uncertainty for user modeling. To this end, we have proposed novel robust formulations for factorization machines and field-aware factorization machines. The distributed Spark based implementation for RFM and RFFM seamlessly scales to massive real-world datasets. Experimental evidence establishes a consistently superior noise resilience of the proposed formulations. This opens up new avenues for utilizing the combined power of robust optimization and traditional factorization machine models. As a future work, benchmarking the effectiveness of RFMs and RFFMs across a breadth of classification settings is a promising area of investigation. Exploring other choices of uncertainty models, like ellipsoidal and conic models is another interesting research direction. Applying RO principles to tree based ensembles and deep learning is yet another unexplored territory. The pursuit of the question of whether incorporating the notion of worst case loss minimization for these highly expressive models results in higher generalization power, might reveal deeper insights about the models themselves.

REFERENCES

[1] Adroll. 2016. *Factorization Machines.* http://tech.adroll.com/blog/data-science/2015/08/25/factorization-machines.html

[2] Dimitris Bertsimas, David B. Brown, and Constantine Caramanis. 2011. Theory and Applications of Robust Optimization. *SIAM Rev.* 53, 3 (Aug. 2011), 464–501. https://doi.org/10.1137/080734510

[3] Alexey Borisov, Ilya Markov, Maarten de Rijke, and Pavel Serdyukov. 2016. A Neural Click Model for Web Search. In *Proceedings of the 25th International Conference on World Wide Web (WWW '16)*. International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, Switzerland, 531–541. https://doi.org/10.1145/2872427.2883033

[4] Olivier Chapelle, Eren Manavoglu, and Romer Rosales. 2014. Simple and Scalable Response Prediction for Display Advertising. *ACM Trans. Intell. Syst. Technol.* 5, 4, Article 61 (Dec. 2014), 34 pages. https://doi.org/10.1145/2532128

[5] Comscore. 2015. *Lessons Learned: Maximizing Returns with Digital Media.* https://www.comscore.com/Insights/Presentations-and-Whitepapers/2015/Lessons-Learned-Maximizing-Returns-with-Digital-Media

[6] Criteo. 2016. *The State of Cross-Device Commerce.* http://www.criteo.com/resources/cross-device-commerce-report-h2-2016/

[7] Laurent El Ghaoui, Gert R. G. Lanckriet, and Georges Natsoulis. 2003. *Robust Classification with Interval Data.* Technical Report UCB/CSD-03-1279. EECS Department, University of California, Berkeley. http://www2.eecs.berkeley.edu/Pubs/TechRpts/2003/5772.html

[8] Emarketer. 2015. *When Will Mobile Marketers Move Beyond Basic Measurement?* https://www.emarketer.com/Article/Will-Mobile-Marketers-Move-Beyond-Basic-Measurement/1012600

[9] He Xinran et. al. 2014. Practical Lessons from Predicting Clicks on Ads at Facebook. In *Proceedings of the Eighth International Workshop on Data Mining for Online Advertising (ADKDD'14)*. ACM, New York, NY, USA, Article 5, 9 pages. https://doi.org/10.1145/2648584.2648589

[10] McMahan et. al. 2013. Ad Click Prediction: A View from the Trenches. In *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '13)*. ACM, New York, NY, USA, 1222–1230. https://doi.org/10.1145/2487575.2488200

[11] Bram L. Gorissen, İhsan Yanikoğlu, and Dick den Hertog. 2015. A practical guide to robust optimization. *Omega* 53, Supplement C (2015), 124 – 137. https://doi.org/10.1016/j.omega.2014.12.006

[12] Yuchin Juan, Damien Lefortier, and Olivier Chapelle. 2017. Field-aware Factorization Machines in a Real-world Online Advertising System. In *Proceedings of the 26th International Conference on World Wide Web Companion (WWW '17 Companion)*. International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, Switzerland, 680–688. https://doi.org/10.1145/3041021.3054185

[13] Yuchin Juan, Yong Zhuang, Wei-Sheng Chin, and Chih-Jen Lin. 2016. Field-aware Factorization Machines for CTR Prediction. In *Proceedings of the 10th ACM Conference on Recommender Systems (RecSys '16)*. ACM, New York, NY, USA, 43–50. https://doi.org/10.1145/2959100.2959134

[14] Sungchul Kim, Nikhil Kini, Jay Pujara, Eunyee Koh, and Lise Getoor. 2017. Probabilistic Visitor Stitching on Cross-Device Web Logs. In *Proceedings of the 26th International Conference on World Wide Web (WWW '17)*. International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, Switzerland, 1581–1589. https://doi.org/10.1145/3038912.3052711

[15] Kevin J. Lang, Benjamin Moseley, and Sergei Vassilvitskii. 2012. Handling Forecast Errors While Bidding for Display Advertising. In *Proceedings of the 21st International Conference on World Wide Web (WWW '12)*. ACM, New York, NY, USA, 371–380. https://doi.org/10.1145/2187836.2187887

[16] Hoai An Le Thi, Xuan Thanh Vo, and Tao Pham Dinh. 2013. *Robust Feature Selection for SVMs under Uncertain Data.* Springer Berlin Heidelberg, Berlin, Heidelberg, 151–165. https://doi.org/10.1007/978-3-642-39736-3_12

[17] Ryan McDonald, Keith Hall, and Gideon Mann. 2010. Distributed Training Strategies for the Structured Perceptron. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics (HLT '10)*. Association for Computational Linguistics, Stroudsburg, PA, USA, 456–464. http://dl.acm.org/citation.cfm?id=1857999.1858068

[18] Mahdi Pakdaman Naeini and Gregory F. Cooper. 2016. Binary Classifier Calibration Using an Ensemble of Near Isotonic Regression Models. *2016 IEEE 16th International Conference on Data Mining (ICDM)* (2016), 360–369.

[19] Z. Pan, E. Chen, Q. Liu, T. Xu, H. Ma, and H. Lin. 2016. Sparse Factorization Machines for Click-through Rate Prediction. In *2016 IEEE 16th International Conference on Data Mining (ICDM)*. 400–409. https://doi.org/10.1109/ICDM.2016.0051

[20] A. D. Pozzolo, O. Caelen, R. A. Johnson, and G. Bontempi. 2015. Calibrating Probability with Undersampling for Unbalanced Classification. In *2015 IEEE Symposium Series on Computational Intelligence*. 159–166. https://doi.org/10.1109/SSCI.2015.33

[21] Y. Qu, H. Cai, K. Ren, W. Zhang, Y. Yu, Y. Wen, and J. Wang. 2016. Product-Based Neural Networks for User Response Prediction. In *2016 IEEE 16th International Conference on Data Mining (ICDM)*. 1149–1154. https://doi.org/10.1109/ICDM.2016.0151

[22] Steffen Rendle. 2010. Factorization Machines. In *Proceedings of the 2010 IEEE International Conference on Data Mining (ICDM '10)*. IEEE Computer Society, Washington, DC, USA, 995–1000. https://doi.org/10.1109/ICDM.2010.127

[23] Christopher Riederer, Yunsung Kim, Augustin Chaintreau, Nitish Korula, and Silvio Lattanzi. 2016. Linking Users Across Domains with Location Data: Theory and Validation. In *Proceedings of the 25th International Conference on World Wide Web (WWW '16)*. International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, Switzerland, 707–719. https://doi.org/10.1145/2872427.2883002

[24] V. Tuzlukov. 2002. *Signal Processing Noise.* CRC Press. https://books.google.co.in/books?id=x6hoBG_MAYIC

[25] Jun Wang, Weinan Zhang, and Shuai Yuan. 2016. Display Advertising with Real-Time Bidding (RTB) and Behavioural Targeting. *CoRR* abs/1610.03013 (2016). arXiv:1610.03013 http://arxiv.org/abs/1610.03013

[26] Matei Zaharia, Reynold S. Xin, Patrick Wendell, Tathagata Das, Michael Armbrust, Ankur Dave, Xiangrui Meng, Josh Rosen, Shivaram Venkataraman, Michael J. Franklin, Ali Ghodsi, Joseph Gonzalez, Scott Shenker, and Ion Stoica. 2016. Apache Spark: A Unified Engine for Big Data Processing. *Commun. ACM* 59, 11 (Oct. 2016), 56–65. https://doi.org/10.1145/2934664

[27] Qian Zhao, Yue Shi, and Liangjie Hong. 2017. GB-CENT: Gradient Boosted Categorical Embedding and Numerical Trees. In *Proceedings of the 26th International Conference on World Wide Web (WWW '17)*. International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, Switzerland, 1311–1319. https://doi.org/10.1145/3038912.3052668

[28] Martin Zinkevich, Markus Weimer, Lihong Li, and Alex J. Smola. 2010. Parallelized Stochastic Gradient Descent. In *Advances in Neural Information Processing Systems 23*, J. D. Lafferty, C. K. I. Williams, J. Shawe-Taylor, R. S. Zemel, and A. Culotta (Eds.). Curran Associates, Inc., 2595–2603. http://papers.nips.cc/paper/4006-parallelized-stochastic-gradient-descent.pdf