Detection of Review Abuse via Semi-Supervised Binary Multi-Target Tensor Decomposition

Anil R. Yelundur
India ML - Amazon
yelundur@amazon.com

Vineet Chaoji
India ML - Amazon
vchaoji@amazon.com

Bamdev Mishra
Microsoft India
bamdevm@microsoft.com

ABSTRACT

Product reviews and ratings on e-commerce websites provide customers with detailed insights about various aspects of the product such as quality, usefulness, etc. Since they influence customers' buying decisions, product reviews have become a fertile ground for abuse by sellers (colluding with reviewers) to promote their own products or to tarnish the reputation of competitor's products. In this paper, our focus is on detecting such abusive entities (both sellers and reviewers) by applying tensor decomposition on the product reviews data. While tensor decomposition is mostly unsupervised, we formulate our problem as a semi-supervised binary multi-target tensor decomposition, to take advantage of currently known abusive entities. We empirically show that our multi-target semi-supervised model achieves higher precision and recall in detecting abusive entities as compared to unsupervised techniques. Finally, we show that our proposed stochastic partial natural gradient inference for our model empirically achieves faster convergence than stochastic gradient and Online-EM with sufficient statistics.

KEYWORDS

E-commerce, Reviews Data, Sellers, Reviewers, Tensor Decomposition, Stochastic Partial Natural Gradients

ACM Reference Format:
Anil R. Yelundur, Vineet Chaoji, and Bamdev Mishra. 2019. Detection of Review Abuse via Semi-Supervised Binary Multi-Target Tensor Decomposition. In *The 25th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '19), August 4–8, 2019, Anchorage, AK, USA.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3292500.3330678

1 INTRODUCTION

Product reviews and ratings on e-commerce websites provide customers with detailed insights about various aspects of the product. Ratings allow customers to gauge the quality of the product as perceived by other customers who have bought the product. Consequently, customers rely significantly on product reviews and ratings while making buying decisions on e-commerce platforms. Given their influence on customer spends, product reviews are a fertile ground for abuse.

A common form of abuse involves a seller running campaigns soliciting fake, genuine-looking positive reviews, for their own products or fake negative reviews about their competitors' products. The paid reviewers have a varied modi operandi. They can either create their own account and start posting paid reviews (fake reviews) or they can hijack an inactive account in good standing to post seemingly innocuous reviews from that account. There are also businesses/agencies which promise a fee for writing fake reviews on Amazon. Figure 1 shows a social media snippet (name anonymized) of a seller or an agency soliciting reviewers for a fee. To circumvent identification, paid reviewers also distribute the volume of fake reviews across multiple accounts.

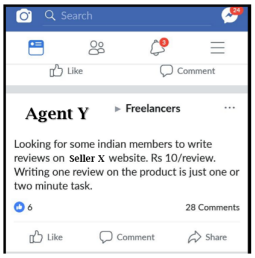


Figure 1: Seller/agency soliciting fake reviewers.

In order to maintain customer's trust on the reviews and in turn on the e-commerce platform, it is imperative for e-commerce websites to ensure that the reviews remain sacrosanct. As a result, identifying and taking enforcement actions on fake reviews, paid reviewers and the underlying abusive sellers is a significant focus area within e-commerce companies. In this paper, we broadly focus on the problem of detecting abusive entities (sellers and reviewers) within the product reviews' ecosystem. We formalize the interactions between sellers, products and reviewers as a binary tensor. Tensors are multidimensional arrays [18] and are used in capturing multidimensional features effectively. We subsequently apply tensor decomposition techniques to identify the dense cores. These dense cores are treated as anomalous interactions within a tensor with predominantly uniformly distributed interactions. Additionally, we use known abusive sellers and reviewers from the past as partial supervision to further enhance the model, with each form of abuse as a separate target signal. To summarize, our technical contributions are as follows:

(1) We formulate detection of abusive entities (sellers and reviewers), a task of identifying dense cores in the seller-reviewer bipartite graph, as a tensor decomposition problem.
(2) We apply unsupervised Bayesian binary tensor decomposition to detect dense blocks in the seller-reviewer relationship. This is based on the Logistic CP tensor decomposition model.
(3) We then develop **se**mi-supervised bi**n**ary mul**ti**-target extension to the u**n**supervised mod**el** (via Pólya-Gamma data

KDD '19, August 4–8, 2019, Anchorage, AK, USA

ACM ISBN 978-1-4503-6201-6/19/08.
https://doi.org/10.1145/3292500.3330678

augmentation) so that we can incorporate prior information about multiple forms of abuse to improve detection of abusive entities. We call our proposed approach ***SENTINEL***.

(4) Finally, we develop stochastic partial natural gradient learning for the semi-supervised model and show that it empirically achieves faster convergence than stochastic gradient descent and EM with sufficient statistics.

(5) We show the efficacy of the proposed approach as compared to the state-of-the-art techniques for tensor decomposition and review abuse detection.

To the best of our knowledge, this is the first time a) semi-supervised multi-target binary tensor decomposition has been applied to the problem of detecting fake entities in the review spam domain, b) natural gradients has been used for inference within tensor decomposition. Although the paper focuses on the problem formulation and scientific aspects of SENTINEL, we have additionally developed an extensive platform that uses machine learning models to flag abusive entities. The platform allows a combination of automated as well as manual enforcement. The feedback from the enforcement gets channeled back into the platform to further enhance SENTINEL.

The rest of this paper is organized as follows. Section 2 introduces related works as well as some background regarding our application of tensor decomposition for detecting abuse in the seller-reviewer relationship. Section 3 describes the baseline unsupervised binary tensor decomposition technique. Section 4 describes SENTINEL, encapsulating the semi-supervised multi-target extensions. Section 5 describes our proposed stochastic partial natural gradient learning for the inference of all the latent parameters of the semi-supervised model. Experimental results are shown in Section 6.

Additional details related to the derivations of natural gradients are in the longer version of this paper [33].

2 RELATED WORK AND BACKGROUND

There has been a lot of attention recently to address the issue of finding fake reviewers in online e-commerce platforms. Jindal et al. [17] were one of the first to show that review spam exists and proposed simple text based features to classify fake reviewers.

Identifying Abusive Reviews: Abusive reviewers have grown in sophistication ever since the initial efforts [16, 17], employing professional writing skills to avoid detection via text-based techniques. In [8], the authors have proposed stylistic features derived from the Probabilistic Context Free Grammar parse trees to detect review spam. To detect more complex fake review patterns, researchers have proposed 1) graph based approaches such as approximate bipartite cores and lockstep behavior detection among reviewers [2, 11, 15, 20], 2) techniques to identify network footprints of reviewers in the reviewer product graph [31], and 3) using anomalies in ratings distribution [10]. Some recent research has pointed at the importance of time in identifying fake reviews since it is critical to produce as many reviews as possible in a short period of time to be economically viable. Methods exploiting temporal and spatial features related to reviewers/reviews [19, 32], as well as the sequence of reviews [21] have been proposed. While it is not possible to capture all the work on review spam detection, [6] provides a broad coverage of efforts in this area.

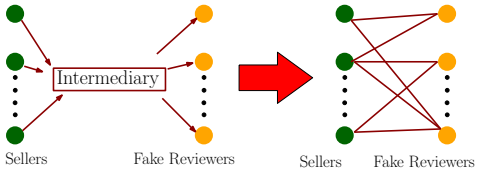


Figure 2: Seller-Reviewer Abuse: key signals.

Tensor based methods: Techniques such as CrossSpot [14], M-Zoom [30], and MultiAspectForensics [23] propose identifying dense blocks in tensors or dense sub-graphs in heterogeneous networks, which can also be applied to the problem of identifying fake reviewers. M-Zoom is an improved version of CrossSpot that computes dense blocks in tensors which indicate anomalous or fraudulent behavior. The number of dense blocks (i.e., sub-tensors) returned by M-Zoom is configured a-priori. Note that the dense blocks identified may be overlapping, i.e., a tuple could be included in two or more blocks. M-Zoom is used as one of the baseline unsupervised methods in our experiments. MultiAspectForensics, on the other hand, automatically detects and visualizes novel patterns that include bipartite cores in heterogeneous networks.

Tensor decomposition [12, 26–29] is also applied to detect abusive entities (i.e., sellers and reviewers) since it facilitates the detection of dense bipartite sub-graphs. Dense bipartite sub-graphs are indicative of suspicious behavior due to the following reason: at any given time, there is a common pool of fake reviewers (paid reviewers) that are available and who are willing to write a positive or negative review in exchange for a fee. Sellers recruit these fake reviewers through an intermediary channel (such as social media groups, third party brokers, etc.) as shown in Figure 2, where nodes on the left indicate sellers and nodes on the right indicate reviewers and an edge indicates a written review. Note that a seller has to recruit a *sizeable* number of fake reviewers in a short amount of time to make an impact on the overall product rating – positive impact for her own products and negative impact for her competitor's products. Given a common pool of available fake reviewers at any given time, suspicious behavior manifests as dense bipartite connections between a group of sellers and a group of reviewers with similar ratings, such as near 5-star or near 1-star ratings. Hence the presence of a bipartite core (a dense bipartite sub-graph) in some contiguous time interval Δt is a strong indicator of abuse by the group of sellers and reviewers involved.

Therefore, our goal boils down to finding bipartite cores (or dense blocks) using the factor matrices resulting from tensor decomposition. The modes of the tensor for our problem space correspond to the seller, reviewer, product, rating, and time of review. The entities in the corresponding factor matrices that have relatively higher values are indicative of abuse. By aggregating these abusive entities across the modes of the tensor results in discovering bipartite cores, where each core consists of a group of reviewers that provide similar rating across a similar set of sellers (and their products) where all of these reviews are occurring within a short contiguous interval of time.

We have a small set of known abusive sellers and reviewers identified via manual audits. We leverage the partial supervision

to propose a semi-supervised extension to tensor decomposition to detect new abusive entities with greater fidelity. To leverage correlations between different forms of abuse (e.g., paid reviews abuse, abuse related to compromised accounts), we incorporate multiple binary targets based on Logistic Model with Pólya-Gamma data augmentation.

Natural gradient learning: Natural gradient learning [1] is an alternative to traditional gradient descent based learning. We develop stochastic partial natural gradient learning for the semi-supervised tensor decomposition model and show that it empirically achieves faster convergence as compared to stochastic gradient descent and EM with sufficient statistics.

3 LOGISTIC CP DECOMPOSITION

Let $\mathcal{X}$ be a 3-mode (aka 3-way) tensor. CP decomposition of a tensor is defined as:

$$\mathcal{X} = \sum_{r=1}^{R} \lambda_r \vec{a}_r \odot \vec{b}_r \odot \vec{c}_r, \tag{1}$$

where λ_r denotes the weight, $\vec{a}_r$, $\vec{b}_r$, and $\vec{c}_r$ are vectors (or rank-1 tensors) and $\odot$ represents vector outer product. R is called the rank of the tensor. *CP Decomposition* is a generalization of matrix decomposition. Each tuple in Amazon's review data captures the 'reviewing/rating' event wherein a reviewer writes a review/rating at a specific time for a product sold by a seller. These relationships can be described as a 5-way (or 4-way, if the seller is omitted) binary tensor. Such a binary tensor is incomplete since not all reviewers would have rated all products. Hence, the elements of a binary tensor can be modeled as being generated from a Logistic function, i.e., a value of *1* when the relationship exists and value of *0* where relationships do not exist. In our probabilistic tensor decomposition framework, we only consider those entries where a current relationship exists.

Logistic CP tensor decomposition is tensor CP decomposition in a probabilistic framework [26–28]. The generative model for a K-mode tensor denoted by $\mathcal{X}$ is:

$$\begin{aligned}
&\mathcal{X} \sim f(\textstyle\sum_{r=1}^{R} \lambda_r \vec{u}_r^{(1)} \odot \cdots \odot \vec{u}_r^{(K)}),\\
&\delta_l \sim \text{Inv-Gamma}(a_l, 1) : a_1 = 1, a_l = a_1 + (l-1)\tfrac{1}{R},\\
&\tau_r = \textstyle\prod_{l=1}^{r} \delta_l,\\
&\lambda_r \sim \mathcal{N}(0, \tau_r),\\
&\boxed{\mu_{i_k,r}^{(k)} \sim \text{Inv-Gamma}(a_c, b_2) : \textbf{Our Enhancement},}\\
&u_{i_k,r}^{(k)} \sim \mathcal{N}(0, \mu_{i_k,r}^{(k)}),
\end{aligned}$$

where f specifies the Bernoulli-Logistic function for the binary valued tensor. Let $\vec{u}_{i_k}^{(k)}$ denote the R dimensional factor corresponding to entity i_k in mode k. The number of non-zero values of $\vec{\lambda}$ determines the *rank* of the tensor. Gaussian priors are assigned to the latent variables $\vec{\lambda}$ and $\vec{u}_{i_k}^{(k)}$ for $k \in [1, K]$. The variance of the Gaussian prior for $\vec{\lambda}$ is controlled by a *Multiplicative Inverse Gamma Process* [7] that has the property of reducing the variance as r increases. To get closed-form updates for all the latent variables, [26–28] introduce additional variables, denoted by ω_i corresponding to each tuple i in the binary tensor that are Pólya-Gamma distributed.

We assign an Inverse-Gamma hyper-prior to the variance of the Gaussian priors of $\vec{u}_{i_k}^{(k)}$. The reason for adding the Inverse-Gamma hyper-prior to the model is that it provides adaptive L2-regularization. Hence, we can control the amount of regularization, i.e., provide different amounts of regularization to factors of the mode(s) that have target information than those without target information.

We use this unsupervised model as our base and develop multi-target semi-supervised extensions to it as described in Section 4. [26–28] propose using either sufficient statistics (in batch EM or as online EM) or Gibbs sampling (in batch) for inference. They claim that the online EM reaches reasonably good quality solutions fairly quickly as compared to their batch counterparts in most situations. However, the online EM does scalar updates of each latent variable that is inherently a vector of dimension R, and this may result in slower convergence. Alternatively, we propose using partial natural gradient learning in a stochastic setting for inference that is both, online in nature as well as allows vectorized updates for each of the latent variables.

4 PROPOSED MULTI-TARGET SEMI-SUPERVISED LOGISTIC CP DECOMPOSITION

In this section, we describe SENTINEL, the semi-supervised extensions to the Logistic CP model for detecting abusive sellers and reviewers. We have prior information associated with a subset of the entities for at least one of the modes. This prior information is specified as binary target(s), where each target corresponds to a specific type of abuse. The framework is called *semi-supervised tensor decomposition* as the tensor decomposition is achieved by incorporating this prior data. The intuition behind using the target information is that the patterns hidden in the known abusive entities could be leveraged to discover, with greater precision, additional entities that have similar signatures. We know that abusive behavioral patterns change as the entities continuously try to game the system. We hypothesize that the proposed semi-supervised framework offers a way to detect such changing behavior earlier and with greater efficiency as compared with unsupervised techniques.

In multi-target semi-supervised tensor decomposition, one or more targets are specified for at least one of the modes (denoted as $\vec{z}_l^{(k)}$ for target l in mode k). We drop the superscript (k) for convenience. This is represented as a matrix where each column corresponds to a single target. For each target:
1) Both positive and negative labels need to be specified.
2) Data can be specified for only a subset of the entities in that mode (semi-supervised learning).

The decomposition of the tensor is achieved by taking the multi-target label information across the mode(s) into account. The label information corresponding to each target is predicted via its own Logistic model whose coefficients are denoted as $\vec{\beta}_l^{(k)}$ for target l in mode k. We drop the superscript (k) for convenience. For a given mode, the covariates of each Logistic model are the same, i.e., the factors corresponding to the entities in that mode. The coefficients $\vec{\beta}_l$ are assigned *Gaussian* priors. Note that the Logistic

model corresponding to each target is learnt by taking into account the correlations of the binary labels from other targets, and therefore called as *Multi-target Learning*. For each Logistic model, to get closed form updates, we introduce additional variables denoted by v_l that are *Pólya-Gamma* distributed. Equation 8 in Appendix (Section A.2) shows how the Logistic likelihood for each task l becomes a quadratic function in $\vec{\beta}_l$ by augmenting the data with v_l.

Multi-Target Learning

Consider binary target information for mode k that is specified as an $M \times L$ matrix, where M denotes the number of entities for which the targets are specified and L denotes the number of tasks. Note that in the semi-supervised setting the binary target information is usually specified for only a subset of the entities of some mode k. Based on the specified target information, the tensor decomposition technique can infer the neighbors of these abusive entities. For example, if a seller a is flagged as having review abuse, then the tensor decomposition technique would infer its factor as a function of the reviewers associated with this seller during some time Δt where the density is high. This in turn would lead to detection of other sellers who are also connected to some subset of these same reviewers during this same time interval. These other sellers would then end up having a similar factor representation indicating a high probability of being abusive.

Given L tasks (where $L > 1$), the learning falls under the multi-target learning framework. Multi-target learning takes into account the correlations (or a similarities) between the L tasks to learn L different classifiers [13]. This is achieved via defining a $L \times L$ matrix where each element l, j of that matrix, denoted by $Q_{l,j}$, carry the reverse cosine information between the targets l and j and is computed as:

$$Q_{l,j} = 1 - \frac{\vec{z}_l^{\top} \vec{z}_j}{|\vec{z}_l|.|\vec{z}_j|}.$$

The interpretation of this $L \times L$ matrix denoted by $\boldsymbol{Q}$ is easily understood by taking a single task l as an example. Note that all the diagonal elements of $\boldsymbol{Q}$ are zero (by definition). For task l, compute the following quantity given below:

$$e^{-\sum_{j \neq l} Q_{l,j} (\vec{\beta}_l{}^{\top} \vec{\beta}_j)}. \tag{2}$$

Equation 2 can be regarded as an exponential distribution on $\vec{\beta}_l$ (without the normalizing factor) with a rate parameter equal to $\sum_{j \neq l} Q_{l,j} . \vec{\beta}_j$. We want to find a suitable $\vec{\beta}_l$ given the rate parameter such that it maximizes this exponential distribution. This is described considering the three extreme cases:

1) If target l, j are negatively-correlated (cosine measure near -1 and $Q_{l,j} \sim 2$), then the coefficients defining their respective classifiers need to be of opposite signs and similar magnitude. This implies that the dot product $\vec{\beta}_l{}^{\top} \vec{\beta}_j$ is negative. And since $Q_{l,j}$ is positive, implies that the exponent is positive.

2) If target l, j are uncorrelated (cosine measure near 0 and $Q_{l,j} \sim 1$), then the coefficients defining their respective classifiers need to be distinct as well. This implies that the dot product $\vec{\beta}_l{}^{\top} \vec{\beta}_j$ will be closer to zero, meaning that the exponent is close to zero.

3) If target l, j are highly correlated (cosine measure near 1 and $Q_{l,j} \sim 0$), then the coefficients defining their respective classifiers need to be very similar (same sign and similar magnitude). This implies that the dot product $\vec{\beta}_l{}^{\top} \vec{\beta}_j$ will be positive. However, since $Q_{l,j} \sim 0$, implies that the exponent will be close to zero.

To handle cases when the magnitude any of element(s) $\vec{\beta}_j$ are very small or very large, we modify equation 2 as given below:

$$e^{-\sum_{j \neq l} Q_{l,j} [\vec{\beta}_l{}^{\top} Sign(\vec{\beta}_j)]}. \tag{3}$$

Taking the natural logarithm of equation 3 yields:

$$-\sum_{j \neq l} Q_{l,j} [\vec{\beta}_l{}^{\top} Sign(\vec{\beta}_j)].$$

The Appendix (sections A.2 and A.3) describes the modeling of the L classifier models in a multi-target framework jointly with the factors from the Logistic CP decomposition. Section A.1 in the Appendix describes the modeling of the latent parameter $\vec{\lambda}$. Next section describes our proposed partial natural gradient inference for the multi-target semi-supervised Logistic CP model.

5 PARTIAL NATURAL GRADIENTS: INFERENCE

Natural gradient is defined as the product of the Euclidean, i.e., standard gradient and the inverse of the Fisher information matrix. Natural gradient learning in the context of online learning is explained in [1]. It is an optimization method that is traditionally motivated from the perspective of information geometry and works well for many applications as an alternate to stochastic gradient descent [22]. Natural gradient descent is generally applicable to the optimization of probabilistic models. It has been shown that in many applications, natural gradients seem to require far fewer total iterations than gradient descent, hence making it a potentially attractive alternate method. However it has been known that for models with many parameters, computing the natural gradient is impractical since it requires computing the inverse of a large matrix, i.e., the Fisher information matrix. This problem has been addressed in prior works where an approximation to the Fisher is calculated such that it is easier to store and invert than the exact Fisher.

The latent variables in our semi-supervised model that we need to infer are $\vec{\lambda}$ of length R, matrix $\boldsymbol{U}^{(k)}$ for each mode $k \in [1, K]$ of dimension $n_k \times R$ and $\vec{\beta}_l^{(k)}$ of length $R + 1$ for target $l \in [1 : L]$ in mode $k \in [1, K]$. The corresponding log-posteriors of these latent variables that we need to maximize are denoted by $\log[g(\vec{\lambda})]$ (defined in A.1), $H(\vec{u}^{(k)}_{i_k=n})$ (defined in A.3) and $F(\vec{\beta}_l^{(k)})$ (defined in A.2), respectively. Natural gradient update in iteration t is then defined as:

$$\begin{aligned}
\vec{\lambda}_{(t)} &= \vec{\lambda}_{(t-1)} + \gamma_t . \mathcal{I}(.)^{-1} . \mathbb{E}\Big[\nabla_{\vec{\lambda}} \log[g(\vec{\lambda})]\Big] \\
\vec{u}^{(k)}_{i_k=n,(t)} &= \vec{u}^{(k)}_{i_k=n,(t-1)} + \gamma_t . \mathcal{I}(.)^{-1} . \mathbb{E}\Big[\nabla_{\vec{u}^{(k)}_{i_k=n}} H(\vec{u}^{(k)}_{i_k=n})\Big] \\
\vec{\beta}^{(k)}_{l,(t)} &= \vec{\beta}^{(k)}_{l,(t-1)} + \gamma_t . \mathcal{I}(.)^{-1} . \mathbb{E}\Big[\nabla^{(k)}_{\vec{\beta}_l} F(\vec{\beta}_l^{(k)})\Big],
\end{aligned} \tag{4}$$

where the learning rate γ_t in (4) is given by:

$$\gamma_t = \frac{1}{(\tau_p + t)^{\theta}}.$$

Note that θ is the forgetting rate and τ_p is the delay. The values for these are chosen such that $\sum_t \gamma_t^2$ is bounded but $\sum_t \gamma_t$ is unbounded.

$\mathcal{I}(.)^{-1}$ in (4) indicates the inversion of the Fisher information matrix (square matrix) in each iteration. There are two difficulties:
1) Computation of the Fisher information matrix is usually not trivial since it may not lend itself in a nice closed form.
2) Size of the Fisher information matrix in our data could possibly be in the tens of thousands or more. This impacts scalability, i.e., could result in very expensive computations that might also pose numerical stability issues leading to an intractable inverse computation.

For the first issue, we show that the Pólya-Gamma data augmentation facilitates easy computation of the Fisher information matrix. The detailed derivations of the partial Fisher information matrices as well as the gradients computations for each of the arguments, namely, $\vec{\lambda}$, $\vec{u}_1^{(1)} \dots \vec{u}_{n_K}^{(K)}$ and $\vec{\beta}_l^{(1)} \cdots \vec{\beta}_L^{(K)}$ are in the longer version of this paper [33].

We have addressed the second issue by exploiting the problem structure which facilitates working with partial Fisher information matrix (hence called partial natural gradients). Partial natural gradients implies that we only work with diagonal blocks of the Fisher information matrix instead of the full matrix. Note that in our problem structure, the loss function is quadratic in each of the arguments $(\vec{\lambda}, \vec{u}_1^{(1)}, \dots, \vec{u}_{n_K}^{(K)}, \vec{\beta}_l^{(1)}, \dots, \vec{\beta}_L^{(K)})$. Due to the *individually* quadratic nature of the loss functions, each diagonal block is a symmetric positive definite matrix of size $R \times R$ for $\vec{\lambda}$ and $\vec{u}_{n_k}^{(k)}$ or $(R+1) \times (R+1)$ for $\vec{\beta}_l^{(k)}$. Hence, the basic convergence guarantees for the full natural gradient learning extends to the partial set up as well [4]. We note that computation of the partial Fisher information matrix is theoretically and numerically tractable as we are dealing with square matrices of size R or $(R+1)$, which is very small (value less than 10) in our problem space.

For scalability over very large data sets, typical of Amazon data, we have implemented partial natural gradient learning in a stochastic setting. A concern here is that in each iteration, using a mini-batch, we are obtaining a noisy estimate of the partial Fisher information matrix. A noisy estimate of this matrix leads to a biased inverse since the mean of inverses is not equal to the inverse of the mean. Hence, theoretical guarantees regarding faster convergence for partial natural gradients in a stochastic setting as compared with stochastic gradients cannot be established. Another side-effect of this noisy estimate is that the matrix may be highly ill-conditioned, and hence may lead to numerical stability problems while computing its inverse. To circumvent this issue, we have experimented with the conditioning of this matrix (conditioning implies adding a scaled diagonal matrix) in either of the two ways. We can either use the prior of the corresponding latent variable to scale the identity matrix and add this to the partial Fisher information matrix (denoted as natural gradient 1). Or we can use the diagonal of the partial Fisher information matrix to scale the identity matrix and add this to the partial Fisher information matrix (denoted as natural gradient 2).

So far, we have not come across any efforts that have applied partial natural gradient learning in a stochastic setting for inference in Bayesian CP tensor decomposition. In section 6, we empirically show that in the semi-supervised setting, on the test data, scalar updates of vector parameters in Online-EM lead to very slow convergence and performance is sub-optimal w.r.t. ROC-AUC as compared with the partial natural gradient and stochastic gradient algorithms. We also empirically show that the partial natural gradient algorithm in a stochastic setting achieves faster convergence than stochastic gradient learning in detecting both abusive sellers and reviewers on Amazon data sets.

6 EXPERIMENTAL RESULTS

This section outlines our experiments to validate the efficacy of SENTINEL towards detecting entities promoting review abuse.

Dataset: For all our experiments, we have taken a random sample of products from the Amazon reviews dataset between 2017 and 2018. Not all reviews are associated with the purchase of a product. As a result, considering only those reviews with an associated seller reduces the dataset size by half. From the review data, we construct two datasets and correspondingly two separate binary tensors - 1) *SELLER-TENSOR* captures the existing association of a reviewer r giving a numeric rating n at time t for a product p from a seller s (5-mode tensor), and 2) *NO-SELLER-TENSOR* captures the existing association of a reviewer r giving a numeric rating n at time t for a product p, without the seller information (4-mode tensor). The former is used to detect abusive sellers, while the latter is used to detect abusive reviewers. The modes of the two binary tensor are reviewer ID, product ID, seller ID (if included), numeric rating and time. Note that numeric rating corresponds to an integer between 1 to 5 and that time is converted to a week index.

SELLER-TENSOR consists of 25K reviewers, 475K products and 70K sellers. NO-SELLER-TENSOR consists of 90K reviewers and 1.4M products. SELLER-TENSOR has 1.2M tuples, resulting in an extremely sparse tensor with a density of $1.6 * 10^{-8}$.

Benchmark Methods: To the best of our knowledge, the proposed binary multi-target semi-supervised enhancement for a CP decomposition and its inference using partial natural gradients are novel to this paper. However, tensor decomposition in general has been applied to detect abusive entities in multi-modal data in the past. Hence, we benchmark the performance of our proposed approach with the following tensor based approaches:

BPTF: Bayesian Poisson Tensor Factorization [29] is a Bayesian CP tensor decomposition approach assuming a Poisson likelihood. The authors in [29] have implemented the unsupervised BPTF model using a batch algorithm, but it does not seamlessly lend itself to semi-supervised extensions.

M-Zoom: The authors in [30] have proposed an unsupervised technique to identify dense blocks in tensors or dense sub-graphs in heterogeneous networks, which can also be applied to identify abusive sellers and reviewers.

Unsupervised BNBCP: Beta Negative-Binomial CP decomposition [12] is also a Bayesian CP tensor decomposition approach, assuming a Poisson likelihood. The BNBCP model is a fully conjugate model and inference is done using Variational Bayes (VB). To be able to scale for massive tensors, we have implemented an online VB version using Stochastic Variational Inference (SVI).

Semi-supervised BNBCP: Since the unsupervised BNBCP model can be easily extended to the semi-supervised setting, we implemented a semi-supervised version. Note that this is a single target and not a multi-target enhancement.

Table 1: Detecting abusive sellers: Comparison between benchmark methods. Note that the numbers are *scaled*.

	Method	Precision	Recall	F1 Score	AUC
Un-Supervised	M-Zoom [30]	**0.61**	0.74	**0.67**	-
	BPTF [29]	0.53	0.79	0.63	0.74
	BNBCP [12]	0.51	**0.89**	0.65	**0.74**
Semi-Supervised (until convergence)	BNBCP [*SVI*]	0.85	0.93	0.89	0.87
	Logistic CP [*Stochastic Gradient*]	**0.89**	0.94	**0.91**	**0.88**
	Logistic CP [*Natural Gradient 1*]	0.85	0.93	0.89	0.88
	Logistic CP [*Natural Gradient 2*]	0.84	**0.94**	0.89	0.88
Semi-Supervised (stop at 200 iterations)	Logistic CP [*Stochastic Gradient*]	0.81	0.74	0.77	0.78
	Logistic CP [*Natural Gradient 1*]	**0.83**	**0.89**	**0.86**	**0.87**
	Logistic CP [*Natural Gradient 2*]	0.78	0.81	0.79	0.85

We use precision, recall and AUC against a test set to measure the performance of the above approaches.

Experimental Validation: Our experiments consist of the following empirical evaluations:

1) Comparison between unsupervised and the proposed semi-supervised enhancements in detecting abusive sellers and reviewers.
2) Impact of different inference techniques on the proposed semi-supervised enhancement.
3) Robustness of the proposed multi-target semi-supervised model to variations in the hyper-parameters.
4) Stability of the proposed approach across many runs, owing to the non-convex nature of tensor decomposition.
5) Scalability of the proposed approach.
6) Comparison with baseline models in terms of early detection of abusive reviewers and its impact on customers viewing the reviews.

Within SENTINEL, we have chosen the following values for the learning rate parameters $\tau_p = 256$ and $\theta = 0.61$. We have chosen the following values for the parameters of the *Inverse Gamma* distributions: $a_c = 1$, $b_1 = 0.4$ and $b_2 = 3$. We have set the mini-batch size to 1024 in all our simulations.

6.1 Detecting Abusive Sellers

In this experiment, we take a random sample of sellers who have been flagged, via manual audits, for being guilty of review abuse. These are treated as positively labeled samples. In addition, we have included a random sample of sellers who are currently not flagged for any kind of abuse. These are treated as negatively labeled samples. Both these samples form the training dataset together. We have taken an additional set of around 1% of sellers [1] as test set to measure the performance of different techniques. The test set has a similar distribution as the training set and has been selected from beyond the training time period. Table 1 shows that all four flavors of the semi-supervised tensor decomposition have higher precision, recall and AUC as compared with the unsupervised techniques - indicating that leveraging behavioral patterns from current abusive sellers improves performance and fidelity in identifying new abusive sellers. Given the single type of seller abuse (aka single target); both semi-supervised BNBCP and SENTINEL models have comparable AUC performance. With early stopping of the semi-supervised models (200 iterations), we observe that inference based on natural gradient outperforms stochastic gradient learning, indicating empirically faster convergence of the former.

[1]High confidence abusive sellers are hard to obtain due to business reasons, hence the small test set.

Table 2: Abusive sellers: evidence from review data.

Suspicious Seller	# Suspicious Products	# Suspicious Reviewers	# Reviews	Density
A	75	25	1802	0.96
B	16	9	132	0.91
C	7	19	114	0.86

Table 3: Detecting abusive reviewers: Comparison between benchmark methods. Note that the numbers are *scaled*.

	Method	Precision	Recall	F1 Score	AUC
Unsupervised	M-Zoom [30]	0.53	**0.70**	**0.60**	-
	BPTF [29]	0.42	0.54	0.47	0.51
	BNBCP [12]	**0.59**	0.46	0.52	**0.61**
Semi-Supervised (until convergence)	BNBCP [*SVI*]	0.60	0.83	0.69	0.79
	Logistic CP [*Stochastic Gradient*]	0.60	0.90	0.72	0.85
	Logistic CP [*Natural Gradient 1*]	**0.62**	0.91	**0.74**	**0.85**
	Logistic CP [*Natural Gradient 2*]	0.59	**0.91**	0.72	0.85
Semi-Supervised (stop at 600 iterations)	Logistic CP [*Stochastic Gradient*]	0.56	0.79	0.66	0.72
	Logistic CP [*Natural Gradient 1*]	**0.59**	**0.87**	**0.71**	**0.83**
	Logistic CP [*Natural Gradient 2*]	0.58	0.86	0.69	0.82

Note that M-Zoom does not produce the scores for each suspicious entity and hence the AUC is not calculated. Due to this lack of scores per entity, we cannot rank the suspicious entities to be able to apply different levels of enforcement actions against them. Among the unsupervised methods, BNBCP has the best AUC, which is around 16% lower than the best semi-supervised approach. The best performing unsupervised method in terms of precision is M-Zoom, which is around 46% lower than the best semi-supervised approach.

Table 2 shows evidence from the review data indicating suspicious behavior for a representative set of sellers that was predicted by SENTINEL to be abusive. Density indicates the ratio of bipartite connections (reviews) observed in the data to the maximum number of bipartite connections between products from this seller and suspicious reviewers. These reviewers are most likely abusive since they have written a review for practically all products by the seller. Since paid reviewer abuse mostly manifests off-Amazon (e.g., funds transferred through bank), evidence gathered from review data is indicative but not legally binding to qualify the sellers as fraudulent. Moreover, since enforcement actions on false positives would impact revenue negatively, the thresholds for enforcement are quite stringent. Majority of the sellers are warned based on the model's recommendation.

6.2 Detecting Abusive Reviewers

In this experiment, we have taken a random sample of reviewers, who have been identified to be guilty of review abuse. Among these, roughly 60% belong to paid reviewer abuse category and the remainder belong to the compromised account abuse category. Recall that the compromised account abuse occurs when a good customer's account is taken over by an abusive reviewer to post fake reviews. SENTINEL supports multi-target labels in a semi-supervised setting. Hence, we treat the two forms of abuse as two separate targets in our model. The BNBCP semi-supervised model, on the other hand, does not support multiple targets. Hence there is no differentiation between these two forms of abuse in the BNBCP semi-supervised model. A random sample of about 80% of this data forms the training set and the remaining 20% forms the test set. Table 3 shows

that all the four variants of our semi-supervised tensor decomposition have higher precision, recall, and AUC as compared with the unsupervised techniques. Recall and AUC have shown significant increase between the unsupervised and semi-supervised methods, as compared to the precision. With early stopping (600 iterations), we see that the natural gradient outperforms stochastic gradient learning, indicating empirically faster convergence of the former. At full convergence, multi-target logistic CP semi-supervised model exhibits better performance as compared with the semi-supervised model that is not designed to differentiate between different forms of abuse (i.e., BNBCP model). To test the hypothesis that jointly learning separate models for each abuse type is better than learning a single model for all abuse types, we experimented with using a single target which is the union of the two abuse types in SENTINEL. Results indicated that we see an almost 3% gain in AUC in predicting one of the targets in the multi-target scenario, hence confirming our hypothesis.

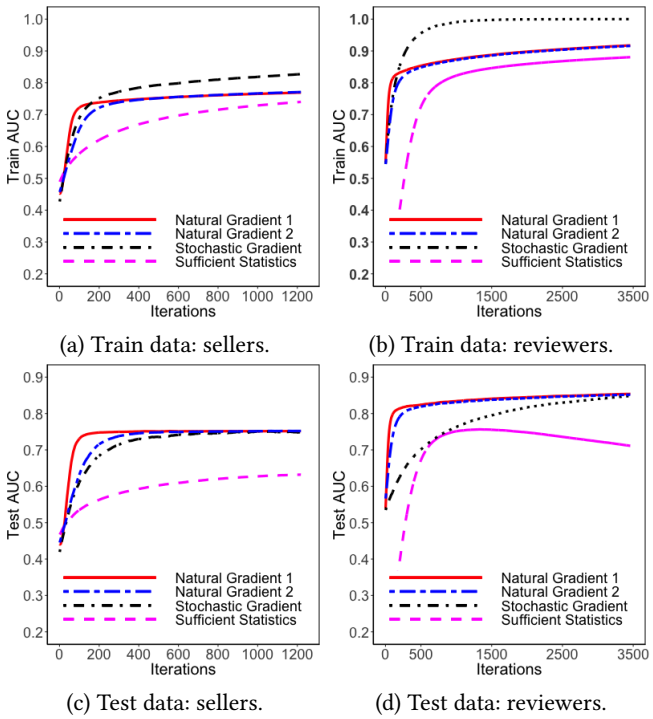
6.3 Stochastic Partial Natural Gradients versus Baseline Learning Methods

We apply multi-target semi-supervised Logistic CP tensor decomposition approach on Amazon review data (SELLER-TENSOR for sellers and NO-SELLER-TENSOR for reviewers) to compare the performance of two baseline learning methods, namely, sufficient statistics (Online-EM) and stochastic gradient with the proposed stochastic partial natural gradient learning. As mentioned in Section 5, we have two flavors of stochastic partial natural gradient learning, differing only by the conditioning applied to the partial Fisher information matrix. Figure 3 shows the ROC-AUC plot for detecting abusive sellers and abusive reviewers versus the iteration number. Solid red plot corresponds to the stochastic partial natural gradient learning type 1, blue (dash-dot) plot corresponds to stochastic partial natural gradient learning type 2, black dashed plot corresponds to stochastic gradient learning and magenta (dash) plot corresponds to online EM with sufficient statistics.

Stochastic gradient learning has a tendency to over-train since it is unable to shrink some of the elements of $\vec{\lambda}$ towards zero as the tensor rank is less than R. Stochastic partial natural gradient learning (both flavors) does not suffer from significant over-training since it is able to shrink 60% of the values of λ_r towards zero within the first one thousand iterations. This leads to similar AUC on train and test data sets for detecting both abusive sellers and abusive reviewers as compared with the other two baselines. Stochastic partial natural gradient learning type 2 has a slightly slower learning rate than type 1. Online-EM with sufficient statistics shows poorer performance on test data (for both reviewers and sellers) when compared with stochastic gradient or partial natural gradient learning.

6.4 Sensitivity to Model Hyper-parameter

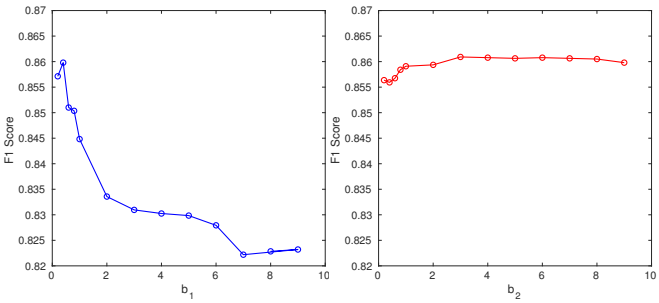
Impact of the scale parameter of the two Inverse-Gamma distributions (i.e., model hyper-parameters), namely, b_1 and b_2 on F1 score in a semi-supervised setting is shown in Figure 4 (*a*) and (*b*). b_1 (b_2) is the scale parameter of the Inverse-Gamma distribution that controls the variance of the Gaussian prior for the factors that belong to the mode(s) associated with (without) multi-target data.



(a) Train data: sellers. (b) Train data: reviewers.

(c) Test data: sellers. (d) Test data: reviewers.

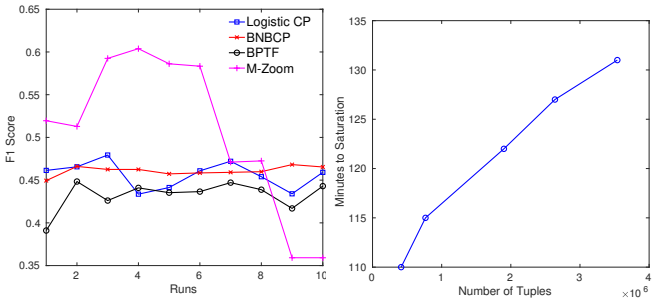
Figure 3: Efficiency of partial natural gradient learning in identifying abusive sellers and reviewers.



(a) F1 Score versus b_1 with b_2 Fixed at 9. (b) F1 Score versus b_2 with b_1 Fixed at 0.4.

Figure 4: Impact of varying hyper-parameters on F1 Score.

b_2 is also the scale parameter for the Inverse-Gamma distribution that controls the variance of the Gaussian prior for the Logistic regression coefficients corresponding to each task l. b_2 is set such that it offers very little regularization, i.e., set to a high value. However, b_1 has to be chosen more carefully such that it provides the right amount of regularization in the semi-supervised setting. In Figure 4 (*a*), we fix b_2 to a arbitrarily high value and vary b_1 from 0.2 until 9. We notice that a value of $b_1 = 0.4$ produces the highest F1 score. In Figure 4 (*b*), we fix $b_1 = 0.4$ and vary b_2 from 0.2 until 9. We notice that the impact on F1 score of varying b_2 is minimal as compared with varying b_1 and that the best F1 score is achieved



(a) Robustness: F1 Scores across Runs.

(b) Scalability: Size of Tensor vs Convergence Time.

Figure 5: Robustness and Scalability.

at $b_2 = 3.0$. Hence all our experiments are done with $b_1 = 0.4$ and $b_2 = 3.0$.

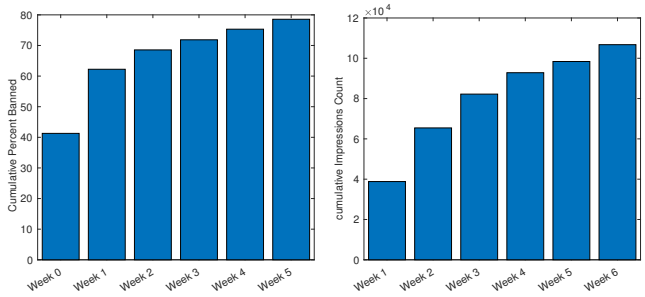
6.5 Robustness of Various Techniques

Due to the non-convex nature of tensor decomposition, Figure 5 (*a*) shows the variations in the F1 scores across ten different runs for each of the techniques, namely, Logistic CP with stochastic partial natural gradient inference, BNBCP, BPTF and M-Zoom (unsupervised setting) for detecting abusive reviewers.

BNBCP has the least variations across different runs, followed by the Logistic CP with stochastic partial natural gradient based inference. M-Zoom (in magenta) requires the number of blocks to be specified as input. Given the number of blocks, it produces identical sub-tensors across different runs. Hence, we have measured the F1 score performance across different number of blocks, monotonically varying them from 5 to 14. Below 5, the F1 score was much lower, hence it is omitted. We see that with 8 blocks as input, the F1 score is at the highest and it beats the other three methods. However, unlike the other three methods, M-Zoom does not produce a score for each suspicious reviewer. That is, we cannot rank the reviewers according to their suspiciousness (that shows that M-Zoom lacks a very important factor for taking enforcement actions).

6.6 Scalability of the proposed approach

We ran the logistic CP with partial natural gradient on the NO-SELLER-TENSOR dataset with varying number of tuples. The hyper-parameters were fixed at the same setting mentioned above. Figure 5 (*b*) shows the scalability of the proposed method across five different datasets of varying sizes. We plot the time to converge (iteration number at which learning is saturated) in minutes versus the size of the tensor (i.e., the number of tuples) in log scale. We notice an almost linear increase in computational time to converge as the number of tuples increases. Our algorithm is also easily parallellizable because in a given iteration, each tuple in the mini-batch can be processed independently. This, we expect should help us in achieving much faster convergence than what we are currently seeing. We are in the process of implementing a parallel version of our algorithm.



(a) Cumulative Percentage of Reviewers Banned versus Time.

(b) Cumulative Impressions Count versus Time.

Figure 6: Impact of Early Detection.

6.7 Impact of Early Detection of Abusive Reviewers

In this section, we present a retrospective analysis of the impact of our model as compared with the existing system.

Early detection of abusive reviewers: Figure 6 (*a*) shows the fraction of suspicious reviewers predicted by SENTINEL, that are later banned by existing models. The model is executed during '$Week_0$' and 40% of the identified suspicious reviewers overlap with those banned by production models. This overlap increases as we move forward by a week at a time, until five weeks later (indicated by *Week 5*) when the overlap is roughly 80%. This indicates that roughly 40% of the reviewers that are detected by our model at *Week 0* are detected across next 5 weeks by the production system. The impact of not banning the abusive reviewers at *Week 0* is captured by the number of impressions created by the reviews from those abusive reviewers until *Week 5*.

Impact in terms of affected impression counts: Impressions implies the number of customers who read the reviews (i.e., fake reviews) from those abusive reviewers until they were banned. Note that banning a reviewer results in removal of all his/her reviews. Hence larger the impression count implies more people have read those reviews which means bigger impact from not banning these abusive reviewers early enough. Figure 6 (*b*) shows the cumulative impact in terms of the number of impressions accumulated over consecutive weeks, from the time they were detected by our model until their banning by existing production model. These impressions on fake reviews quantify the potential impact of our model, once deployed in production.

6.8 Experiments on public data

To allow others to reproduce our results, we applied SENTINEL to the public Amazon Customer Reviews dataset[2] [9]. We selected the product categories *Clothing, Shoes, and Jewelry*, *Home and Kitchen* and *Sports and Outdoors* for this test. Note that this data does not have seller information. Also since this data is from May 1996 to July 2014, we do not have any labelled data, i.e., known abusive reviewers corresponding to that period. To test SENTINEL, we used random 80% of the suspicious reviewers from M-Zoom to seed SENTINEL as well as the semi-supervised version of BNBCP. The remaining 20% of the suspicious reviewers were considered as the

[2]The dataset is available at http://jmcauley.ucsd.edu/data/amazon/.

test set. On the test set, we obtain a 5.6% increase in AUC with SENTINEL as compared with the semi-supervised BNBCP model and 5.9% increase in AUC compared with the best unsupervised model, i.e., BPTF. Since we are considering M-Zoom's output as the ground truth, we acknowledge that this is not an ideal setting to validate the performance of the proposed approach. Having said that, this setup at least allows us to loosely compare various methods.

7 CONCLUSION

We formulated the problem of identifying abusive entities (i.e., sellers and reviewers) as a tensor decomposition problem and proposed semi-supervised enhancements by incorporating binary multi-target information for a subset of entities, i.e., known abusive sellers and/or reviewers. Our results demonstrated that the proposed approach (titled SENTINEL) beats the state-of-the-art baselines in detecting abusive entities.

We have shown, in the supplementary manuscript, that Pólya-Gamma formulation simplifies calculation of the partial Fisher information matrix, and hence proposed a scalable stochastic partial natural gradient learning for inference of all the latent variables of the semi-supervised model. We have empirically shown that our inference using stochastic partial natural gradient learning achieves faster convergence than online EM using sufficient statistics and stochastic gradient learning.

Future Work: Given the equivalence between tensor decomposition and convolutional rectifier networks [5], we would like to compare the performance of the latter (hierarchical Tucker) with our probabilistic CP decomposition model. We also want to investigate the feasibility of applying *Graph Convolutional Networks* to our multi-modal data (with data being either reviewers and products and their relationships w.r.t. to rating as well as time of rating OR reviewers, products and sellers and their relationships) to detect abusive entities.

ACKNOWLEDGMENT

This work was started at Amazon under the guidance of Srinivasan H. Sengamedu who suggested application of semi-supervised tensor decomposition based techniques towards review abuse detection. We would also like to thank Purushottam Kar for sharing helpful insights related to the biased estimates of the partial Fisher information matrices in our stochastic algorithm.

REFERENCES

[1] S. I. Amari. 1998. Natural gradient works efficiently in learning. *Neural computation* 10, 2 (1998), 251–276.
[2] A. Beutal, W. Xu, V. Guruswami, C. Palow, and C. Faloutsos. 2013. CopyCatch: stopping group attacks by spotting lockstep behavior in social networks. In *Proceedings of the 22nd International Conference on World Wide Web (WWW)*. 119–130.
[3] A. Bhattacharya and B. D. David. 2011. Sparse Bayesian infinite factor models. *Biometrika* 98, 2 (2011), 291.
[4] L. Bottou, F. E. Curtis, and J. Nocedal. 2018. Optimization methods for large-scale machine learning. *SIAM Rev.* 2, 60 (2018), 223–311.
[5] N. Cohen, Sharir O., and Shashua A. 2016. On the expressive power of deep learning: a tensor analysis *(JMLR: Workshop and Conference Proceedings)*, Vol. 49. 1–31.
[6] M. Crawford, T. M. Khoshgoftaar, J. D. Prusa, A. N. Richter, and H. Al Najada. 2015. Survey of review spam detection using machine learning techniques. *Journal of Big Data* 2, 1 (05 Oct 2015), 23.
[7] D. Durante. 2016. *A note on the multiplicative gamma process.* Technical Report. arXiv preprint arXiv:1610.03408.
[8] S. Feng, R. Banerjee, and Y. Choi. 2012. Syntactic stylometry for deception detection. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Short Papers - Volume 2 (ACL '12)*. 171–175.
[9] R. He and J. McAuley. 2016. Ups and Downs: modeling the visual evolution of fashion trends with one-class collaborative filtering. 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, 507–517.
[10] B. Hooi, N. Shah, A. Beutal, S. Gunneman, L. Akoglu, M. Kumar, D. Makhija, and C. Faloutsos. 2016. BIRDNEST: Bayesian Inference for ratings-fraud detection. In *SIAM International Conference on Data Mining (SDM)*.
[11] B. Hooi, H. Song, A. Beutal, N. Shah, K. Shin, and C. Faloutsos. 2016. Frauder: Bounding graph fraud in the face of camouflage. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 895–904.
[12] C. Hu, P. Rai, C. Chen, M. Harding, and L. Carin. 2015. Scalable Bayesian non-negative tensor factorization for massive count data. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases (ECML/PKDD)*. 53–70.
[13] J. Huang, G. Li, Q. Huang, and X. Wu. 2015. Learning label specific features for multi-label classification. In *IEEE International Conference on Data Mining (ICDM)*. 181–190.
[14] M. Jiang, A. Beutal, P. Cui, B. Hooi, S. Yang, and C. Faloutsos. 2015. A general suspiciousness metric for dense blocks in multimodal data. In *IEEE International Conference on Data Mining (ICDM)*. 781–786.
[15] M. Jiang, P. Cui, A. Beutal, C. Faloutsos, and S. Yang. 2015. Inferring lockstep behavior from connectivity pattern in large graphs. *Knowledge and Information Systems* 48, 2 (2015), 399–428.
[16] N. Jindal and Liu. B. 2008. Opinion spam and analysis. In *Proceedings of the 2008 International Conference on Web Search and Data Mining*. 219–230.
[17] N. Jindal and B. Liu. 2007. Analyzing and detecting review spam. In *Proceedings of the 2007 Seventh IEEE International Conference on Data Mining (ICDM '07)*. 547–552.
[18] T. G. Kolda and B. W Bader. 2009. Tensor decompositions and applications. *SIAM review* 51, 3 (2009), 455–500.
[19] H. Li, Z. Chen, A. Mukherjee, B. Liu, and J. Shao. 2015. Analyzing and detecting opinion spam on a large-scale dataset via temporal and spatial patterns. In *Proceedings of the 9th International AAAI Conference on Web and Social Media (ICWSM)*. 26–29.
[20] Y. Li, O. Matrinez, X. Chen, and J. E. Hopcroft. 2016. In a world that counts: Clustering and detecting fake social engagement at scale. In *Proceedings of the 25th International Conference on World Wide Web (WWW)*. 111–120.
[21] Yuming Lin, Tao Zhu, Hao Wu, Jingwei Zhang, Xiaoling Wang, and Aoying Zhou. 2014. Towards Online Anti-opinion Spam: Spotting Fake Reviews from the Review Sequence. In *Advances in Social Networks Analysis and Mining (ASONAM)*. 261–264.
[22] J. Martens. 2014. *New insights and perspectives on the natural gradient method.* Technical Report. arXiv preprint arXiv:1412.119.
[23] K. Maruhashi, F. Guo, and C. Faloutsos. 2011. MultiAspectForensics: Pattern mining on large-scale heterogenous networks with tensor analysis.. In *Advances in Social Networks Analysis and Mining (ASONAM)*. 203–210.
[24] J. W. Pillow and J. G. Scott. 2012. Fully Bayesian inference for neural models with negative-binomial spiking. In *Advances in Neural Information Processing Systems (NeurIPS)*.
[25] N. G. Polson, J. G. Scott, and J. Windle. 2013. *Bayesian inference for logistic models using Polya-Gamma latent variables.* Technical Report. arXiv preprint arXiv:1205:0310v3.
[26] P. Rai, C. Hu, M. Harding, and L. Carin. 2015. Scalable probabilistic tensor factorization for binary and count data. In *Proceedings of the 24th International Conference on Artificial Intelligence (IJCAI)*. 3770–3776.
[27] P. Rai, Y. Wang, and L. Carin. 2015. Leveraging features and networks for probabilistic tensor decomposition. In *Association for the Advancement of Artificial Intelligence (AAAI)*. 2942–2948.
[28] P. Rai, Y. Wang, S. Guo, G. Chen, D. Dunson, and L. Carin. 2014. Scalable Bayesian low-rank decomposition of incomplete multiway tensors. In *Proceedings of the 31st International Conference on Machine Learning (ICML)*. 1800–1808.
[29] A. Schein, J. Paisley, D. M. Blei, and H. Wallach. 2015. Bayesian Poisson tensor facorization for inferring multilateral relations from sparse dyadic event counts. In *Proceedings of the 21st ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 1045–1054.
[30] K. Shin, B. Hooi, and C. Faloutsos. 2016. M-zoom: Fast dense-block detection in tensors with quality quarantees.. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases (ECML/PKDD)*. 264–280.
[31] J. Ye and L. Akoglu. 2015. Discovering opinion spammer groups by network footprints. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. 267–282.

[32] J. Ye, S. Kumar, and L. Akoglu. 2016. Temporal opinion spam detection by multivariate indicative signals. In *International AAAI Conference on Web and Social Media*. 743–746.
[33] A. R. Yelundur, V. Chaoji, and B. Mishra. 2019. *Detection of review abuse via semi-supervised binary multi-target tensor decomposition.* Technical Report. arXiv preprint arXiv:1905.06246.

A APPENDIX

A.1 Determine $\vec{\lambda}$

$\vec{\lambda}$ has a Gaussian prior whose variance is determined via Multiplicative Gamma Process [3, 7] aka *Adaptive Dimensionality Reduction* technique that induces sparsity and automatically deletes redundant parameters. The Multiplicative Gamma Process consists of a multiplicative Gamma prior on the precision of a Gaussian distribution that induces a multiplicative Inverse-Gamma prior on its variance parameter. Since its performance is sensitive to the hyper-parameter settings of the Inverse-Gamma prior we follow certain strategies, shown in [7], to set their values. The generative model for λ_r for r in $[1, R]$ and the Multiplicative Gamma Process are:

$$\delta_l \sim \text{Inv-Gamma}(a_l, 1)$$
$$\lambda_r \sim \mathcal{N}(0, \tau_r),$$

where:

$$\tau_r = \prod_{l=1}^{r} \delta_l. \tag{5}$$

The idea is that for increasing r, τ_r should be decreasing in a probabilistic sense. In other words, the following stochastic order should be maintained with high probability, i.e.,

$$\tau_1 \geq \tau_2 \geq \cdots \geq \tau_R.$$

To guarantee such a stochastic order with a high probability, as suggested in [7], we need to set $a_1 > 0$ and $a_r > a_{r-1}$ and non-decreasing for all $r > 1$. Hence, we choose the hyper-parameters values as follows:

$$a_1 = 1$$
$$a_r = a_1 + (r-1).\frac{1}{R}.$$

The update for δ_r in iteration t is given by (see [28] for details):

$$\delta_r = \frac{1 + \sum_{h=r}^{R} \frac{\lambda_h^2}{2} \prod_{l=1, l\neq r}^{h} \frac{1}{\delta_l}}{0.5(R-r+1) + a_r + 1}. \tag{6}$$

From (5), we can calculate the τ_rs in iteration t for $r \in [1, R]$. Denote $\vec{\tau}$ as the vector consisting of τ_r for $r \in [1, R]$.

Let λ_r for $r \in [1, R]$ denote an element of $\vec{\lambda}$. We introduce auxiliary variables (Pólya-Gamma distributed variables [24, 25]), denoted by ω_i for each element i of the input tensor data, via data augmentation technique.

Consider a mini-batch defined at iteration t as I_t. Define for each $i \in I_t$ and $\phi_i = \vec{\lambda}^T A_i$ where: A_i denotes a vector consisting of elements $A_i^r = \prod_{k=1}^{K} u_{i_k,r}^{(k)}$ for $r \in [1, R]$. Let A be the matrix whose rows are A_i for $i \in I_t$. Let $\hat{\omega}_i$ for $i \in I_t$ be the expected value of the auxiliary variable ω_i corresponding to the i^{th} element of the input tensor data. The expected value of ω_i has a closed-form solution given by:

$$E[\omega_i] = \hat{\omega}_i = \frac{\tanh(\frac{\phi_i}{2})}{2\phi_i}.$$

The update for $\vec{\lambda}$ in iteration t, with the current mini-batch I_t, is obtained by maximizing the natural logarithm of the posterior distribution of $\vec{\lambda}$ given by:

$$\log[g(\vec{\lambda})] := \max_{\vec{\lambda}} \Big[\Big[\sum_{i \in I_t} \kappa_i \phi_i - \frac{\omega_i \phi_i^2}{2} \Big] - \Big[\frac{(\vec{\lambda} \oslash \sqrt{\vec{\tau}})^\top (\vec{\lambda} \oslash \sqrt{\vec{\tau}})}{2} \Big] \Big], \tag{7}$$

where $\kappa_i = y_i - \frac{1}{2}$ and $y_i \in \{0, 1\}$. And the operator $\oslash$ represents element-wise division between the two vectors $\vec{\lambda}$ and $\vec{\tau}$. Note that (7) is a quadratic equation in ϕ_i i.e., $\vec{\lambda}$, and hence has a *closed-form* update.

A.2 Determine the coefficients $\vec{\beta}_l^{(k)}$ for $l \in [1, L]$

The generative model for $\beta_{l,r}^{(k)}$ for r in $[0, R]$, l in $[1, L]$ is:

$$\rho_{l,r}^{(k)} \sim \text{Inv-Gamma}(a_c, b_2),$$
$$\beta_{l,r}^{(k)} \sim \mathcal{N}(0, {\rho_{l,r}^{(k)}}^2).$$

For mode k, we drop the notation (k) for all variables. Define for each $i \in M$ where M denotes the number of elements in mode k that have binary side information corresponding to one or more targets. The logistic function i.e., the likelihood $\mathcal{L}_{l,m}$ corresponding to element m with label $z_{l,m}$ is given by:

$$\mathcal{L}_{l,m} = \frac{1}{1 + exp[-z_{l,m} \vec{\beta}_l^\top \tilde{\vec{u}}_m]},$$

where $\tilde{\vec{u}}_m$ denotes $\vec{u}_m$ prepended with 1 to account for the bias. With introduction of Pólya-Gamma variables [25] denoted by $\vec{\nu}_l$ for task l, the likelihood with the data augmentation becomes:

$$\mathcal{L}_{l,m} = exp(z_{l,m}.\frac{\psi_{l,m}}{2} - \hat{\nu}_{l,m}.\frac{\psi_{l,m}^2}{2}),$$

where:

$$\psi_{l,m} = \vec{\beta}_l^\top (\tilde{\vec{u}}_m)$$
$$\mathbb{E}[\nu_{l,m}] = \hat{\nu}_{l,m} = \frac{tanh(\frac{\psi_{l,m}}{2})}{2\psi_{l,m}} \tag{8}$$

The update for $\vec{\beta}_l$ in iteration t is obtained by maximizing the natural logarithm of the posterior distribution of $\vec{\beta}_l$ given by:

$$F(\vec{\beta}_l) := \max_{\vec{\beta}_l} \Big[\sum_{m=1}^{M} [z_{l,m} \frac{\psi_{l,m}}{2} - \nu_{l,m} \frac{\psi_{l,m}^2}{2}] - \frac{(\vec{\beta}_l \oslash \vec{\rho}_l)^\top (\vec{\beta}_l \oslash \vec{\rho}_l)}{2} - \sum_{j \neq l} Q_{l,j} [\vec{\beta}_l^\top Sign(\vec{\beta}_j)] \Big]. \tag{9}$$

Equation (9) is a quadratic equation in $\vec{\beta}_l$ and hence has a *closed-form* update. And the operator $\oslash$ represents element-wise division between the two vectors $\vec{\beta}_l$ and $\vec{\rho}_l$.

Subsequently, the update for $\rho^2_{l,r}$ at time step t is given by:

$$\rho^2_{l,r_{(t)}} = \frac{\beta^2_{l,r_{(t-1)}}}{2a_c+3} + \frac{2b_1}{2a_c+3}. \tag{10}$$

A.3 Determine the factors $\vec{u}^{(k)}_{i_k,r}$ for mode k

The generative model for $u^{(k)}_{i_k,r}$ is:

$$\begin{aligned} u^{(k)}_{i_k,r} &\sim \mathcal{N}(0, \mu^2_{i_k,r}) \\ \mu^2_{i_k,r} &\sim \text{Inv-Gamma}(a_c, b_1) \quad \text{with target information} \\ \mu^2_{i_k,r} &\sim \text{Inv-Gamma}(a_c, b_2) \quad \text{without target information.} \end{aligned}$$

The Inverse-Gamma hyper-prior on the variance parameter of the Gaussian prior for $u^{(k)}_{i_k,r}$ provides *adaptive L2-Regularization*. The Inverse-Gamma parameters are set so that greater amount of regularization is provided for mode k that has target information. Denote $\vec{u}_{k,i_k}$ as the R dimension vector consisting of factors $u^{(k)}_{i_k,r}$ corresponding to element i_k in mode k and $r \in [1,R]$. Define for each $i \in I_t$, $i_k = n$, and $\phi_i = (\vec{u}^{(k)}_{i_k=n})^T \vec{C}_{i_k=n}$, where each element of $\vec{C}_{i_k=n}$ is $C_{i_k=n,r} = \lambda_r . \prod_{k' \neq k}^{K} u^{(k')}_{i_{k'},r}$ for $r \in [1,R]$. Let $\boldsymbol{C}^{(k)}$ be the matrix whose rows are $\vec{C}_{i_k}$.

Mode k without target information: For mode k, we drop the notation (k) for all variables. The update for $\vec{u}_{i_k=n}$ in iteration t is obtained by maximizing the natural logarithm of the posterior distribution of $\vec{u}_{i_k=n}$ given by:

$$H(\vec{u}_{i_k=n}) := \max_{\vec{u}_{i_k=n}} \Big[\sum_{i \in I_t : i_k = n} [\frac{\phi_i}{2} - \frac{\omega_i \phi_i^2}{2}] \\ \frac{(\vec{u}_{i_k=n} \oslash \vec{\mu}_{i_k=n})^\top (\vec{u}_{i_k=n} \oslash \vec{\mu}_{i_k=n})}{2} \Big], \tag{11}$$

where the operator $\oslash$ represents element-wise division between the two vectors $\vec{u}_{i_k=n}$ and $\vec{\mu}_{i_k=n}$.

Equation (11) is a quadratic equation in $\vec{u}_{i_k=n}$ and hence has a *closed-form* update.

Subsequently the update for $\mu^2_{i_k=n,r}$ at time step t is given by:

$$\mu^2_{i_k=n,r_{(t)}} = \frac{u^2_{i_k=n,r_{(t-1)}}}{2a_c+3} + \frac{2b_2}{2a_c+3}. \tag{12}$$

Mode k with binary target information: Given binary target information for mode k we drop the notation (k) for all variables. Let $z_{l,n}$ denote the binary label (either $+1$ or -1) for element n for task l. Let $\nu_{l,n}$ correspond to the data augmented variable that is Pólya-Gamma distributed for element the n and task l.

The update for $\vec{u}_{i_k=n}$ in iteration t is obtained by maximizing the natural logarithm of the posterior distribution of $\vec{u}_{i_k=n}$ given by:

$$H(\vec{u}_{i_k=n}) := \max_{\vec{u}_{i_k=n}} \Big[\sum_{i \in I_t : i_k = n} [\frac{\phi_i}{2} - \frac{\omega_i \phi_i^2}{2}] \\ - \frac{(\vec{u}_{i_k=n} \oslash \vec{\mu}_{i_k=n})^\top (\vec{u}_{i_k=n} \oslash \vec{\mu}_{i_k=n})}{2} \\ + \sum_{l=1}^{L} [z_{l,n} \frac{\vec{u}^\top_{i_k=n} \hat{\vec{\beta}}_l}{2} - \frac{\hat{\nu}_{l,n} [\beta_{0,l} + \vec{u}^\top_{i_k=n} \hat{\vec{\beta}}_l]^2}{2}] \Big], \tag{13}$$

where $\hat{\vec{\beta}}_l$ denotes the vector of R coefficients without the bias. Note that in (13) we are training a Logistic model using the binary target information to detect abusive entities. And the operator $\oslash$ represents element-wise division between the two vectors $\vec{u}_{i_k=n}$ and $\vec{\mu}_{i_k=n}$. Also (13) is a quadratic equation in $\vec{u}_{i_k=n}$ and hence has a *closed-form* update.

Subsequently the update for $\mu^2_{i_k=n,r}$ at time step t is given by:

$$\mu^2_{i_k=n,r_{(t)}} = \frac{u^2_{i_k=n,r_{(t-1)}}}{2a_c+3} + \frac{2b_1}{2a_c+3}. \tag{14}$$

Algorithm

Algorithm 1 presents the pseudo-code for the multi-target semi-supervised CP tensor decomposition using partial natural gradients.

Algorithm 1: Partial Natural Gradient.

1: Randomly initialize $\vec{\tau}$, $\vec{\lambda}$, $\vec{u}^{(1)}_1 \ldots \vec{u}^{(K)}_{n_K}$ and $\vec{\beta}^{(1)}_l \cdots \vec{\beta}^{(K)}_L$.
2: Set the step-size schedule γ_t appropriately.
3: **repeat**
  a: Sample (with replacement) mini-batch I_t from the training data.
  b: For $i \in I_t$ set
    – $A^r_i = \prod_{k=1}^{K} u^{(k)}_{i_k,r}$ for $r \in [1, R]$
    – $\hat{\omega}_i = \frac{\tanh(\frac{\phi_i}{2})}{2\phi_i}$ where $\phi_i = \vec{\lambda}^\top A_i$
    – $N_{ii} = \frac{1}{\left[\exp[-\frac{\phi_i}{2}] + \exp[\frac{\phi_i}{2}]\right]^2}$.
  c: For $k \in [1, K]$ and $l \in [1, L]$ (where applicable) set
    – $C^{(k)}_{i_k=n,r} = \lambda_r \prod_{k' \neq k}^{K} u^{(k')}_{i_{k'},r}$ for $r \in [1, R]$
    – For entity $i_k = m$ with binary target information $z^{(k)}_{l,m}$:
      $\hat{\nu}^{(k)}_{l,m} = \frac{\tanh(\frac{\psi^{(k)}_{l,m}}{2})}{2\psi^{(k)}_{l,m}}$ where $\psi^{(k)}_{l,m} = \vec{\beta}_l{}^{(k)\top} \tilde{\vec{u}}_{i_k=m}$
      $O_{l:m=n} = \frac{1}{\left[\exp[-\frac{\psi_{l,m=n}}{2}] + \exp[\frac{\psi_{l,m=n}}{2}]\right]^2}$
    – Compute *gradient* and *partial Fisher information matrix* w.r.t. $\vec{\beta}^{(k)}_l$ and update $\vec{\beta}^{(k)}_l$ using (4).
    – Update $\rho_{l,r}{}^{(k)^2}$ using (10).
    – Compute *gradient* and *partial Fisher information matrix* w.r.t. $\vec{u}^{(k)}_{i_k:i \in I_t}$ and update $\vec{u}^{(k)}_{i_k:i \in I_t}$ using (4).
    – Update $\mu^2_{i_k:i \in I_t,r}$ using (12) or (14).
  d: Compute *gradient* and *partial Fisher information matrix* w.r.t. $\vec{\lambda}$ and update $\vec{\lambda}$ using (4).
  e: Update $\vec{\tau}$ using (5).
4: **until** forever