E-commerce Anomaly Detection

A Bayesian Semi-Supervised Tensor Decomposition Approach using Natural Gradients

Anil R Yelundur Bamdev Mishra

Amazon India Development Center, Bangalore, India

December 28, 2018

Agenda

- Problem Setting
- Our Approach & Contributions
- Bayesian Model: Semi-Supervised Binary Tensor CP Decomposition
- Model Inference Techniques
- Partial Natural Gradient Learning
- Experimental Results
- Summary & Possible Future Work

Anomalies: Seller Incentivization of Reviewers

- Fake reviews are a major trust buster for Amazon.
- Agencies solicit people to write reviews on Amazon for a fee.
- Sellers incentivize reviewers to fake good reviews about own products and fake bad reviews about competitors.
    - Increase the rating of the sellers own product(s) and decrease the rating of the product(s) from his competitors

Figure: Facebook snippet showing seller/agency soliciting & incentivizing fake reviewers.

Anomaly Detection: Key Signals

- Entities soliciting fake reviews form dense bipartite cores with their fake customers.
- Fake reviews have similar ratings.
- Fake reviews are temporally clustered.

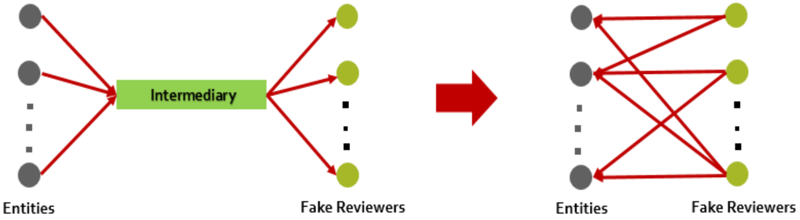


Figure: Dense bi-partite cores between entities & fake reviewers.

Our Approach & Contributions

- Represent anomalies as dense cores in the seller-reviewer bipartite graph satisfying *lockstep* behavior.
    - Input is a tensor because incorporates review timestamp, rating, product & feature information.
- Apply un-supervised Bayesian binary tensor decomposition to detect dense cores between sellers & reviewers.
- Develop semi-supervised Bayesian tensor decomposition to leverage partially labelled data.
    - Labelled data i.e., binary targets – associated with a subset of entities (known abusive sellers and/or reviewers).
- Develop partial natural gradient learning for inference of all the latent variables in the model.

Bayesian CP Factorization – Generative Model

$$\mathcal{Y} \sim f(\sum_{r=1}^{R} \lambda_r \vec{u}_r^{(1)} \odot \cdots \odot \vec{u}_r^{(K)}) : \text{Bernoulli-Logistic function for binary tensor}$$

$$\delta_l \sim \text{Inv-Gamma}(a_l, 1) \text{ and } \tau_r = \prod_{l=1}^{r} \delta_l : a_1 = 1, a_l = a_1 + (l-1)\frac{1}{R}$$

$$\lambda_r \sim \mathcal{N}(0, \tau_r)$$

$$\mu_{i_k,r}^{(k)} \sim \text{Inv-Gamma}(1, 9) : \textbf{Our Enhancement}$$

$$u_{i_k,r}^{(k)} \sim \mathcal{N}(0, \mu_{i_k,r}^{(k)}).$$

Refer to ICML 2014 paper titled *Scalable Bayesian Low-Rank Decomposition of Incomplete Multiway Tensors* by Piyush Rai et al. for details

Semi-Supervised Enhancement to Bayesian CP Factorization

- Currently known abusive entities (a subset of sellers and/or buyers) are represented as binary targets - with label $+1$.
- Tensor decomposition is achieved by simultaneously incorporating the binary target information
  - Requires both positive & negative labels - hence semi-supervised
  - Intuition - patterns hidden in the known abusive entities are leveraged to discover more entities with similar signatures.
- Let $z_n^{(k)}$ denote the label ($+1$ or -1) for element n in mode k.

Refer to our arXiv paper *https://arxiv.org/abs/1804.03836* for details

Semi-Supervised Enhancement to Bayesian CP Factorization

- Let $\hat{\boldsymbol{\beta}}^{(\boldsymbol{k})}$ denote the vector of R coefficients for mode k target.
  - Let $\boldsymbol{\beta}^{(\boldsymbol{k})}$ denote the vector of $R+1$ coefficients for mode k target - including the bias denoted as $\beta_0^{(k)}$.
  - Coefficients $\boldsymbol{\beta}^{(\boldsymbol{k})}$ are assigned Gaussian priors.
- Let $\boldsymbol{u}_{\boldsymbol{n}}^{(\boldsymbol{k})}$ denote the R dimensional vector of factors for element n in mode k that has a prior label associated with it.
- Formulation:
  $P(z_n^{(k)} = 1) = \text{Logistic}(\beta_0^{(k)} + \hat{\boldsymbol{\beta}}^{(\boldsymbol{k})\top} \boldsymbol{u}_{\boldsymbol{n}}^{(\boldsymbol{k})})$.

Semi-Supervised Enhancement to Bayesian CP Factorization

- To get closed form updates of coefficients and factors:
  - Auxiliary variables (Pólya-Gamma distributed) denoted by $\nu_n^{(k)}$ are introduced for each element n in mode k that has a prior label associated with it.
- Let M denote the number of elements in mode k that have binary labels. Then:
- Let $\tilde{\vec{u}}_{i_k=m}$ denote $\vec{u}_{i_k=m}$ prepended with 1, to account for the bias.

Semi-Supervised Enhancement to Bayesian CP Factorization

- The logistic function (likelihood) $\mathcal{L}_m^{(k)}$ corresponding to element m with label $z_m^{(k)}$ is given by:

$$\mathcal{L}_m^{(k)} = \frac{1}{1 + exp[-z_m^{(k)} \vec{\beta}^{(k)}{}^{\top} \tilde{\vec{u}}_{i_k=m}]}.$$

- With introduction of $\vec{\nu}^{(k)}$; the joint likelihood corresponding to element m with label $z_m^{(k)}$ becomes:

$$\mathcal{L}_m^{(k)} = exp(z_m^{(k)} \frac{\psi_m^{(k)}}{2} - \nu_m^{(k)} \frac{{\psi_m{}^{(k)}}^2}{2}),$$

where:

$$\psi_m^{(k)} = \vec{\beta}^{(k)}{}^{\top} \tilde{\vec{u}}_{i_k=m}$$

Model Inference Techniques

- Gibbs Sampling
- Online EM using Sufficient Statistics
- Stochastic Gradient Ascent
- **Natural Gradient Ascent**

.. natural gradient descent makes more progress per step than gradient descent because it implicitly uses a local quadratic model/approximation of the objective function which is more accurate (any much less conservative) than the one implicitly used by gradient descent.

Refer to arXiv 2017 Paper titled *New insights and perspectives on the natural gradient method* by James Martens for details

What is Natural Gradient

The usual definition of the natural gradient (Amari, 1998) which appears in the literature is

$$\tilde{\nabla} h = F^{-1} \nabla h \,,$$

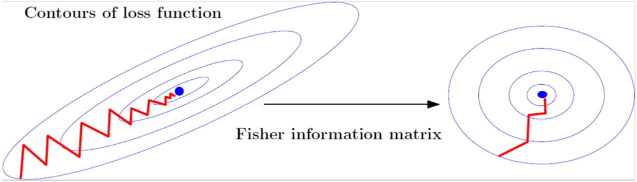


Figure: Natural Gradient Definition & Graphical Representation.

Refer to 1998 Neural Computation Paper titled *Natural gradient works efficiently in learning* by Shun-ichi Amari for details

Challenges in Applying Natural Gradient Learning to our Problem

- Requires inversion of a square matrix of size in the millions in each iteration: computationally very expensive and could pose numerical stability issues
- To circumvent this; we exploit the problem structure:
  - Loss function is quadratic in each of the arguments $(\vec{\lambda}, \boldsymbol{U}, \boldsymbol{\beta})$
- Leads to a simpler approximation of the Fisher information matrix: it facilitates working with the partial block structure of the Fisher information matrix
  - Block approximation has a positive definite structure, implying the basic convergence guarantees for the full natural gradient learning extends to the partial set up
- Computations of the approximate Fisher information matrix is theoretically and numerically tractable

Partial Natural Gradient Learning

- For scalability: implemented Stochastic **Partial Natural Gradient Ascent** updates where a small mini-batch of B samples is chosen in each iteration.
- The partial Hessian structure of our problem is highlighted in the figure below.
  - We propose computationally efficient partial natural learning algorithm, at each iteration the parameters are updated along a stochastic gradient direction, which is scaled with the inverse partial Fisher information matrix.

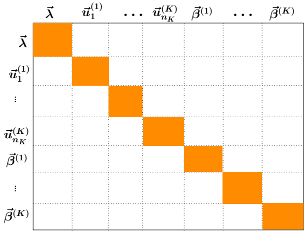


Figure: The block diagonal terms in the Fisher information matrix are *strictly* positive definite and are computationally easy to invert.

Computation of Fisher Information Matrix: Basics

$$F = \mathrm{E}_{Q_x}\left[\mathrm{E}_{P_{y|x}}\left[\nabla \log p(y|x,\theta) \nabla \log p(y|x,\theta)^\top\right]\right] \quad \text{or} \quad F = -\mathrm{E}_{Q_x}\left[\mathrm{E}_{P_{y|x}}\left[H_{\log p(y|x,\theta)}\right]\right]$$

Since $Q(x)$ is usually not available; we replace $Q(x)$ with $\hat{Q}(x)$ as:

$$F = \frac{1}{|S|}\sum_{x \in S} \mathrm{E}_{P_{y|x}}\left[\nabla \log p(y|x,\theta) \nabla \log p(y|x,\theta)^\top\right] \quad \text{or} \quad F = -\frac{1}{|S|}\sum_{x \in S} \mathrm{E}_{P_{y|x}}\left[H_{\log p(y|x,\theta)}\right]$$

Figure: Computation of Fisher Information Matrix.

Refer to arXiv 2017 Paper titled *New insights and perspectives on the natural gradient method* by James Martens for details

Computation of Fisher Information Matrix

Consider the exponent of the function to be maximized w.r.t. $\vec{\lambda}$:

$$g(\vec{\lambda}) := \exp\Big[\sum_{i\in I_t} \kappa_i\phi_i - \frac{\omega_i\phi_i^2}{2}\Big] \exp\Big[-\frac{(\vec{\lambda} \oslash \sqrt{\vec{\tau}})^\top(\vec{\lambda} \oslash \sqrt{\vec{\tau}})}{2}\Big], \quad (1)$$

where

$$\kappa_i = y_i - \frac{1}{2} \text{ and } \phi_i = \vec{\lambda}^\top A_i.$$

- First term in RHS of (1) is the joint conditional likelihood $\mathcal{L}$ of:
    - The binary outcome $y_i \in \{0, 1\}$ denoted as $P(y_i|\vec{\lambda}, A_i)$
    - And the Pólya-Gamma distributed variable denoted as $P(\omega_i|\vec{\lambda}, A_i)$.
- Second term in RHS of (1) is the *Gaussian* prior on $\vec{\lambda}$ with variance $\vec{\tau}$.

Computation of Fisher Information Matrix: Continued

- The joint conditional likelihood term is un-normalized; hence we do the following:
  - Compute the partial Fisher information matrix only w.r.t. the data y_i
  - Marginalize over the data augmented variable ω_i and use the identity:

$$\frac{\exp[\phi_i]^{y_i}}{1+\exp[\phi_i]} = \frac{\exp[\kappa_i \phi_i]}{2} \int_0^\infty \exp[-\frac{\omega_i \phi_i^2}{2}] p(\omega_i) \mathrm{d}\omega_i.$$

- Results in a closed-form, denoted by $\mathcal{L}_i$, which is a normalized likelihood:

$$\mathcal{L}_i = \frac{\exp[\phi_i]^{y_i}}{1+\exp[\phi_i]} = \frac{\exp[\kappa_i \phi_i]}{2} \frac{1}{\cosh[\frac{\phi_i}{2}]} = \frac{\exp\left[\kappa_i \phi_i\right]}{\exp\left[-\frac{\phi_i}{2}\right] + \exp\left[\frac{\phi_i}{2}\right]}.$$

Computation of Fisher Information Matrix: Continued

Partial Fisher Information matrix, denoted by $\mathcal{I}_{\mathcal{L}}(\vec{\lambda})$, with respect to $\vec{\lambda}$ for the likelihood $\mathcal{L}$ is:

$$\mathcal{I}_{\mathcal{L}}(\vec{\lambda}) = - \mathop{\mathbb{E}}_{y_i : i \in I_t} \Big[\sum_{i \in I_t} \frac{\partial^2 log[\mathcal{L}_i]}{\partial \vec{\lambda}^2} \Big] = [A_{I_t}{}^{\top} N_{I_t} A_{I_t}],$$

- A_{I_t} denotes the matrix whose rows are A_i for $i \in I_t$
- N_{I_t} denotes the diagonal matrix whose diagonal elements are N_{ii} for $i \in I_t$; where:

$$N_{ii} = \frac{1}{\Big[\exp[-\frac{\phi_i}{2}] + \exp[\frac{\phi_i}{2}] \Big]^2}.$$

Prior term is accounted by considering it's precision as a conditioner, hence:

$$\mathcal{I}_{\mathcal{L}}(\vec{\lambda}) = [A_{I_t}{}^{\top} N_{I_t} A_{I_t}] + \text{diag}[\vec{\tau}]^{-1},$$

where $\text{diag}[\vec{\tau}]^{-1}$ denotes inverse of a diagonal matrix whose diagonal is $\vec{\tau}$.

Experiment 1: Un-supervised vs Semi-supervised

DATA (Amazon Review Data): Contiguous 10 months of data

Target (Semi-Supervised) : Known abusive Sellers

Table 1: Abusive sellers: un-Supervised and semi-supervised results - *relative performance*.

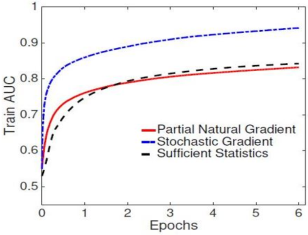
	Method	Precision	Recall	AUC
Un-Supervised	M-Zoom [Shin *et al.*, 2016]	83.8	83.3	-
	BPTF [Schein *et al.*, 2015]	97.2	88.5	90.0
	BNBCP [Hu *et al.*, 2015]	93.3	78.2	78.8
	Logistic CP [*Natural Gradient*]	94.1	68.7	77.6
Semi-Supervised	Logistic CP [*Sufficient Statistics*]	83.3	92.3	91.0
	Logistic CP [*Stochastic Gradient*]	88.9	94.9	92.2
	Logistic CP [*Natural Gradient*]	**100.0**	**100.0**	**100.0**

Figure: Relative Precision, Recall & AUC in Detecting Abusive Sellers.

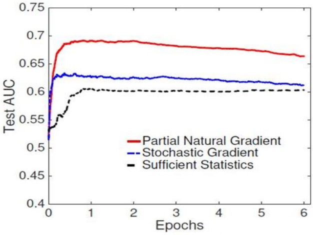
Experiment 2: Natural Gradient vs Stochastic Gradient & Online EM with Sufficient Statistics

DATA (Amazon Review Data): Contiguous 10 months of data

Target (Semi-Supervised) : Known abusive Sellers



(a) Train data: sellers.



(c) Test data: sellers.

Figure: Train & Test AUC Plots.

Summary & Possible Future Work

- We have applied Bayesian binary tensor decomposition to detect dense cores (anomalies) between sellers & reviewers.
- We have shown the application of partial natural gradient learning to infer the latent parameters of the semi-supervised Bayesian CP model.
  - Exploited the quadratic nature of the loss functions to overcome the challenges in applying the full natural gradient learning to our problem.
  - Empirically shown the efficiency of partial natural gradient learning as compared with stochastic gradients and online EM with sufficient statistics.
- Given the equivalence between tensor decomposition and convolutional rectifier networks (*On the Expressive Power of Deep Learning: A Tensor Analysis* arXiv 2016, by Cohen et al.):
  - We could compare the performance of the later with our tensor based anomaly detection.

Questions?