Detecting Fake Review Farms using User-Item Interaction Structure

Priyanshu Agrawal and Niteesh Saravanan

Fraudulent Behaviors

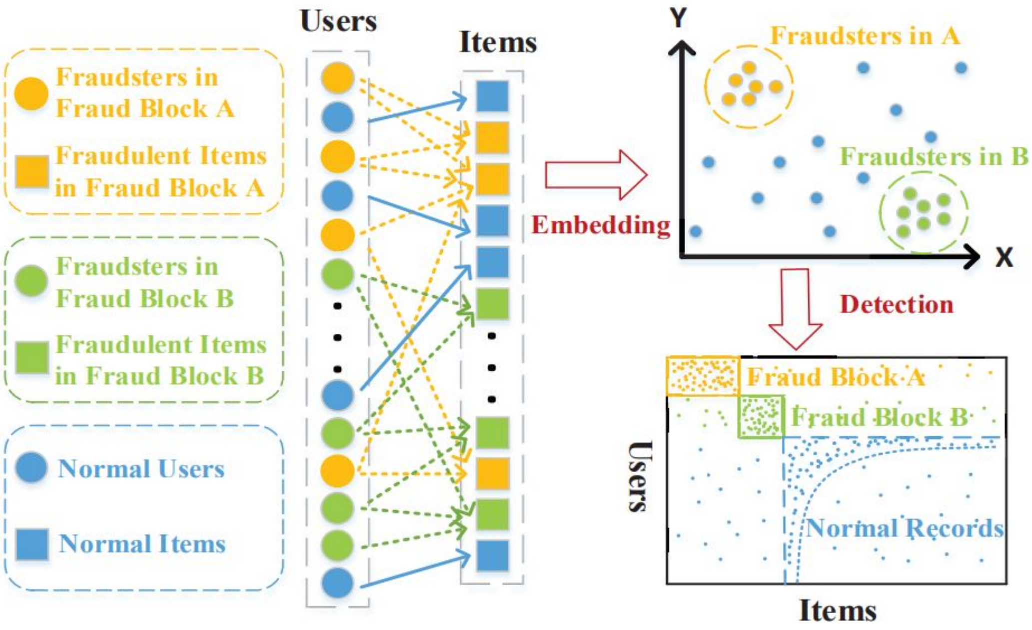
- A typical fraudulent behavior pattern online is fake interactions (reviews or clicks) with items (such as products or websites) to manipulate their metrics.
- Misleads consumers and costs non-fraudulent enterprises.

[1]

[1] B. Jendruszak, “How to spot and prevent fake reviews,” SEON, https://seon.io/resources/how-to-spot-and-prevent-fake-reviews/ (accessed Oct. 13, 2023).

Related Work: DeepFD

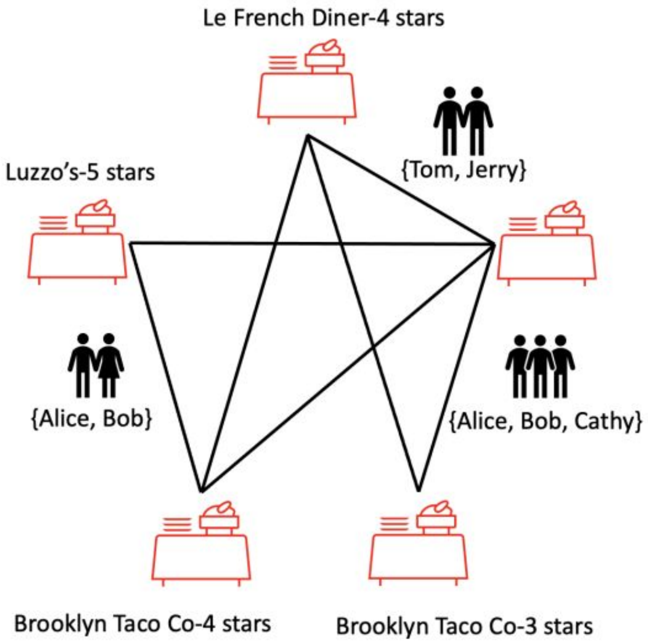
- Models interactions between users and items as bipartite graph.
- Embeds user nodes with an autoencoder and finds dense regions in latent space with DB-SCAN.



[1] H. Wang, C. Zhou, J. Wu, W. Dang, X. Zhu and J. Wang, ``Deep Structure Learning for Fraud Detection," 2018 IEEE International Conference on Data Mining (ICDM), Singapore, 2018, pp. 567-576, doi: 10.1109/ICDM.2018.00072.

Related Work: REAL (modulaRity basEd grAph cLustering)

- Graph nodes represent a specific rating of an item and an edge between two nodes represents all the users who have made both ratings.
- A spectral modularity based graph convolutional network (GCN) generates clusters.
- Clusters are evaluated using a custom defined anomaly score metric which also considers review metadata.



[1]

[1] C. Cao, S. Li, S. Yu and Z. Chen, ``Fake Reviewer Group Detection in Online Review Systems," 2021 International Conference on Data Mining Workshops (ICDMW), Auckland, New Zealand, 2021, pp. 935-942, doi: 10.1109/ICDMW53433.2021.00122.

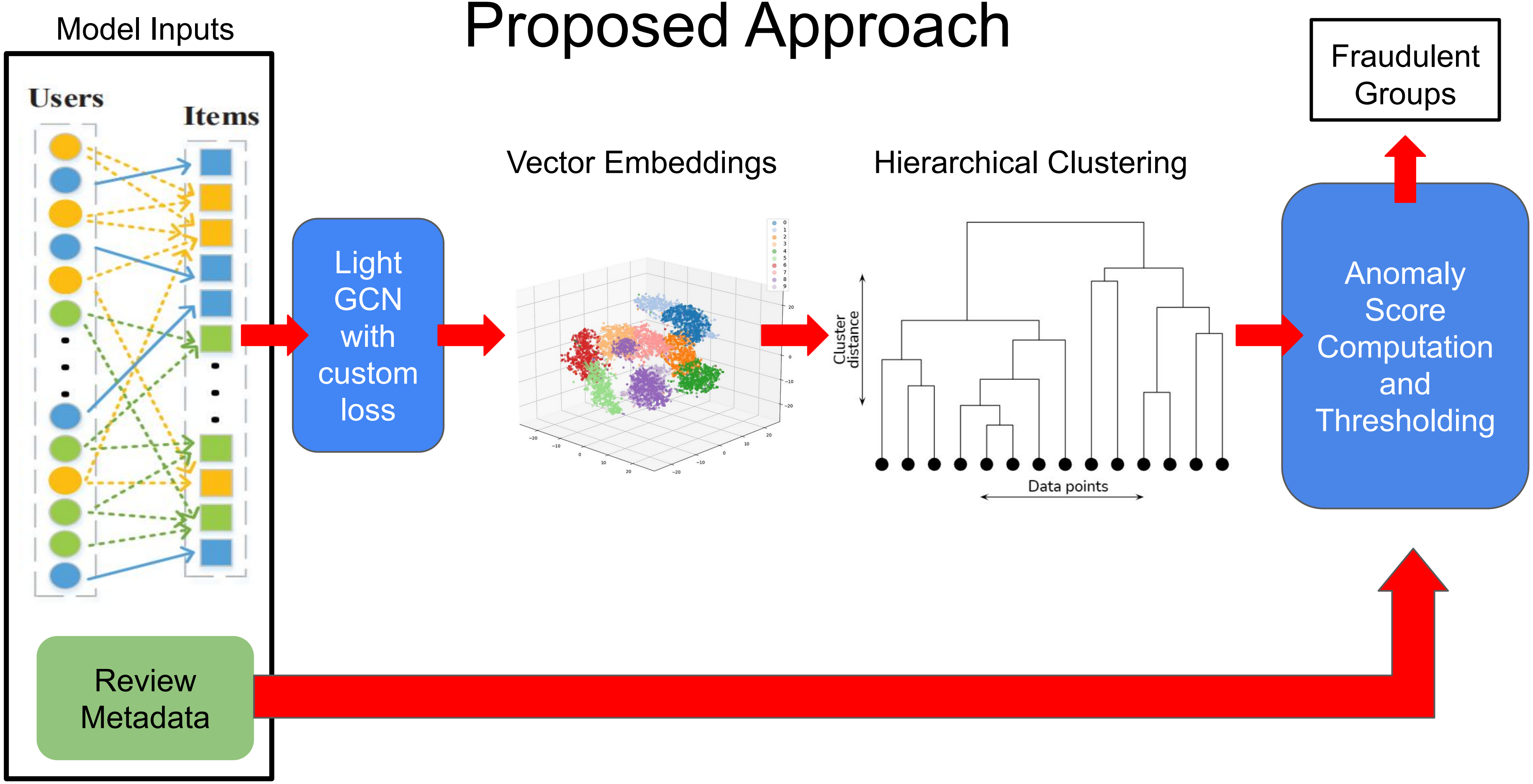
Challenges

- Model should be:
  - Accurate and precise.
  - Efficient and lightweight.
  - Unsupervised (Not reliant on manually labeled training data).
  - Not sensitive to hyperparameters which may hard to choose.
    - DeepFD, which uses DBScan, is highly sensitive to epsilon value.
    - REAL is sensitive to the number of clusters.

- Imperfect datasets:
  - Missing data.
  - Incorrect labels.
  - Class imbalance.

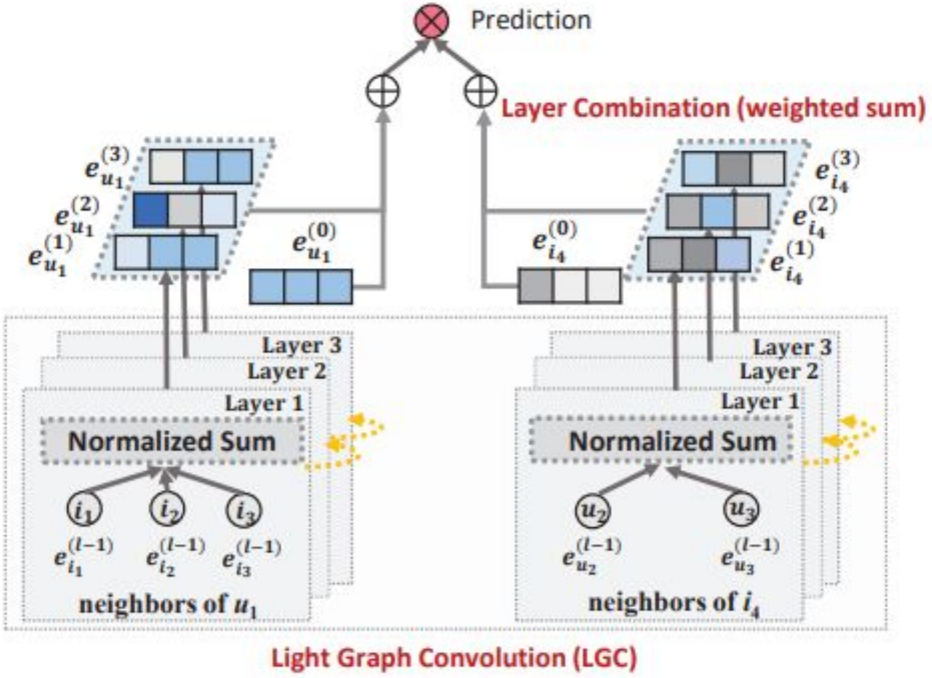
[1] S. Yu, J. Ren, S. Li, M. Naseriparsa, and F. Xia, “Graph learning for fake review detection,” Frontier, 2022, https://www.frontiersin.org/articles/10.3389/frai.2022.922589/.

Proposed Approach



LightGCN Embeddings

- Fast and efficient GCN architecture originally applied to generate embeddings for collaborative filtering.
- We adapt this architecture to our problem by modifying the loss from a BPR loss to a Similarity based loss.



[1]

[1] X. He, K. Deng, X. Wang, Y. Li, Y. Zhang, and M. Wang, ``LightGCN: Simplifying and Powering Graph Convolution Network for Recommendation," Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, 2020.

Embeddings Loss Function

- We use the similarity based loss function used in DeepFD [1].
- The true value of user similarity is jaccard similarity of item sets.
- Similarity in the embeddings is space is exp(-d), where d is the euclidean distance.

$$sim_{ij} = \frac{|N_i \cap N_j|}{|N_i \cup N_j|}$$

N_i is the set of items reviewed by user *i*. N_j is the set of items reviewed by user *j*.

$$\widehat{sim}_{ij} = \exp\left(-\lambda \cdot dis_{ij}\right)$$

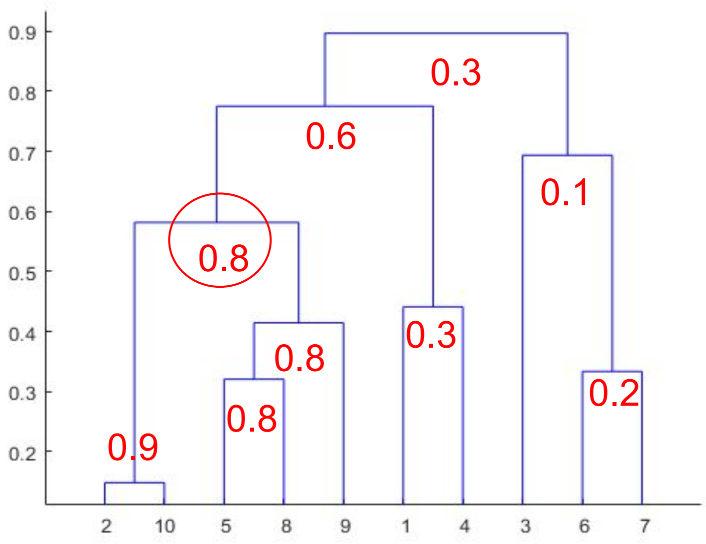
$$\mathcal{L}_{sim} = \sum_{i,j=1}^{m} sim_{ij} \cdot ||\widehat{sim}_{ij} - sim_{ij}||_2^2$$

[1]

[1] H. Wang, C. Zhou, J. Wu, W. Dang, X. Zhu and J. Wang, ``Deep Structure Learning for Fraud Detection," 2018 IEEE International Conference on Data Mining (ICDM), Singapore, 2018, pp. 567-576, doi: 10.1109/ICDM.2018.00072.

Hierarchical Clustering

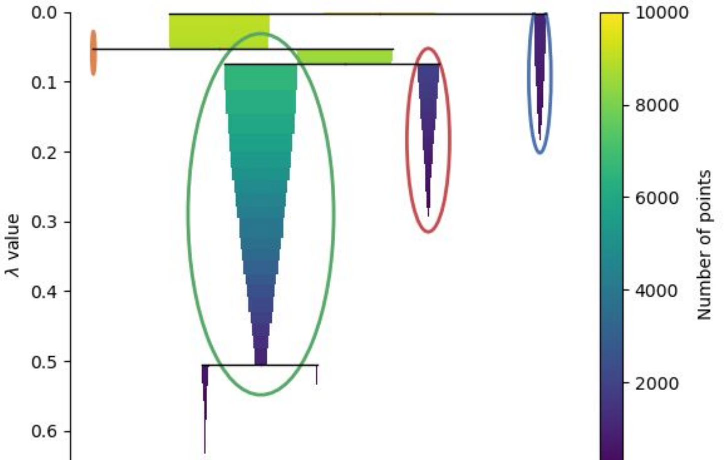
- We would like to avoid too many hyperparameters such as number of clusters or epsilon (density).
- We build a tree of clusters and compute an anomaly score for each node.
- Anomaly score > a threshold is fraud.
- Now we can identify the sub-clusters which are fraudulent without adding additional false positives.



Example hierarchical clustering dendrogram on a fake dataset with 10 points.

Hierarchical DBScan

- For large datasets, performing hierarchical clustering is expensive: $O(n^3)$ time and $O(n^2)$ memory with Ward linkage.
- Therefore, we can instead use HDBScan, which produces a condensed tree in $O(n^2)$ time and $O(n)$ memory.
- Finds clusters of varying density.



Example of an HDBScan condensed tree. Higher lambda values are denser groups.

Anomaly Scores (Tightness)

We used a modified version of the anomaly metric used by REAL [1]. Groups are more likely to be fraudulent if they are tight.

R(g) = set of reviewers in group g

P(g) = set of products reviewed by g

Group Anomaly Compactness

$$\Pi = L(g) * NT(g) * PT(g) * RT(g)$$

Penalty for Small Groups

$$L(g) = \frac{1}{1+e^{-(\beta*(|R(g)|+|P(g)|)-3)}}$$

$\beta = 0.15$ in our experiments

Review Tightness

$$RT(g) = \frac{\sum_{i \in \mathcal{R}(g)} |\mathcal{P}_i|}{|\mathcal{R}(g)||\mathcal{P}(g)|}$$

Product Tightness

$$PT(g) = \frac{|\bigcap_{r \in \mathcal{R}_g} \mathcal{P}_i|}{|\bigcup_{r \in \mathcal{R}_g} \mathcal{P}_i|}$$

Neighbor Tightness

$$NT(g) = \frac{\sum_{i,j \in \mathcal{R}(g)} JS(\mathcal{P}_i, \mathcal{P}_j)}{|\mathcal{R}_g|^2}$$

[1] C. Cao, S. Li, S. Yu and Z. Chen, ``Fake Reviewer Group Detection in Online Review Systems," 2021 International Conference on Data Mining Workshops (ICDMW), Auckland, New Zealand, 2021, pp. 935-942, doi: 10.1109/ICDMW53433.2021.00122.

Anomaly Scores (Metadata)

- For each reviewer in a dataset, we compute ARD and BST. From [1].
- Average Rating Deviation (ARD) is average deviation of a user's star ratings from the average ratings of each product. Normalized to 0-1.
- Burstiness (BST) is larger if the reviewer is active over a shorter period of time.

$$BST(i) = \begin{cases} 0 & if\ E(i) - F(i) > \tau, \\ 1 - \frac{E(i)-F(i)}{\tau} & otherwise \end{cases}$$

Requires hyperparameter τ. We use 30 days.

Final anomaly score, Ω, is a weighted harmonic mean of compactness, average ARD and average BST of a group.

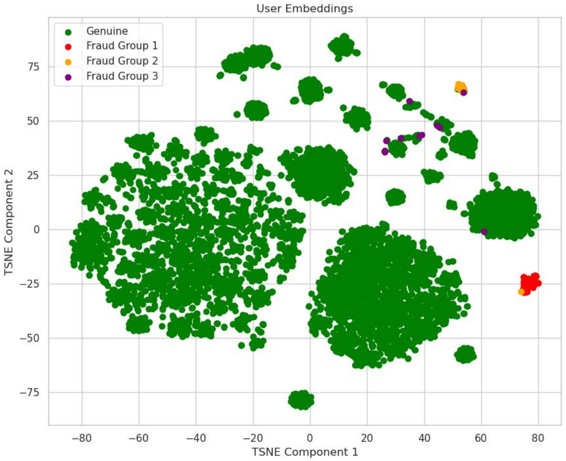
Weights
Compactness: 0.8
Average ARD: 0.1
Average BST: 0.1

[1] C. Cao, S. Li, S. Yu and Z. Chen, ``Fake Reviewer Group Detection in Online Review Systems," 2021 International Conference on Data Mining Workshops (ICDMW), Auckland, New Zealand, 2021, pp. 935-942, doi: 10.1109/ICDMW53433.2021.00122.

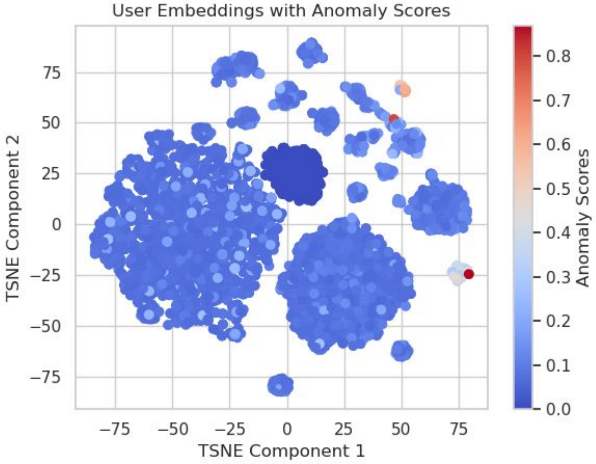
Experimental Visualizations on Small Synthetic Dataset

- For visualizations, we generated a synthetic dataset with 10000 users and 100 products.
- We randomly distributed 30000 reviews across the reviewers and items.
- We picked 3 random groups of users to be fraudulent:
    - Group 1: 100 users, 20 products
    - Group 2: 50 users, 30 products
    - Group 3: 20 users, 10 products
- For fraudulent reviewers, we added additional reviews where every reviewer in a group reviewed the same products.
- Lastly, random noise was added to each fraudulent user. We randomly added and removed one review from each user.

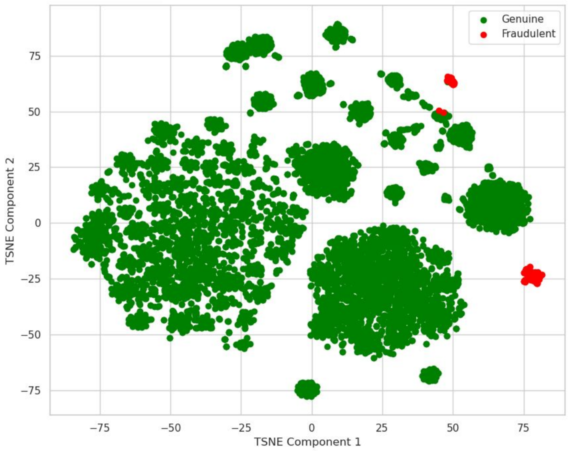
Experimental Visualizations on Small Synthetic Dataset



Embeddings with Ground Truths



Embeddings with Predicted Scores



After Best Threshold (0.3)

99.7% Accuracy

0.985 Precision, 0.808 Recall

Groups 1 and 2 are classified correctly, but Group 3 is obscured by noise

Testing On Real World Dataset

- Main testing is done with the YelpNYC fake review dataset.
- It contains restaurants in New York City.
- It is labeled by Yelp's filters, which are assumed to be near ground truth.

Dataset	#Reviews	#Reviewers	#Products
YelpNYC	359,052	160,225	923

[1]

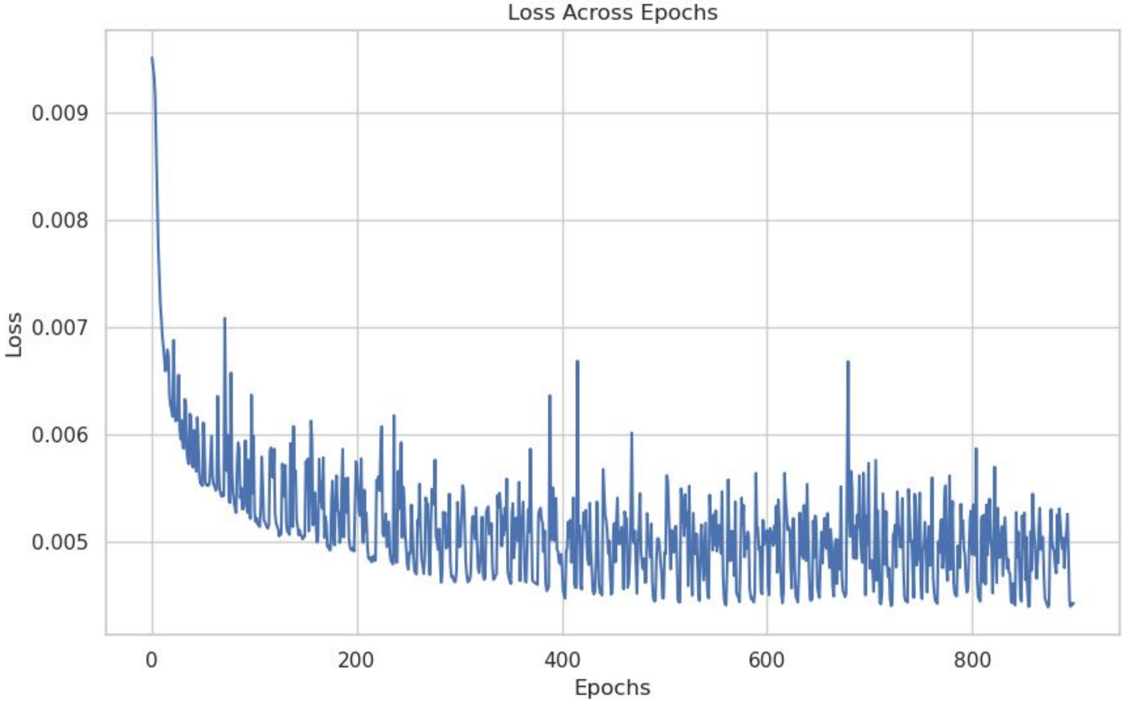
[1] C. Cao, S. Li, S. Yu and Z. Chen, ``Fake Reviewer Group Detection in Online Review Systems," 2021 International Conference on Data Mining Workshops (ICDMW), Auckland, New Zealand, 2021, pp. 935-942, doi: 10.1109/ICDMW53433.2021.00122.

Embedding Time with LightGCN on YelpNYC

YelpNYC dataset with 16 dimensional embeddings.

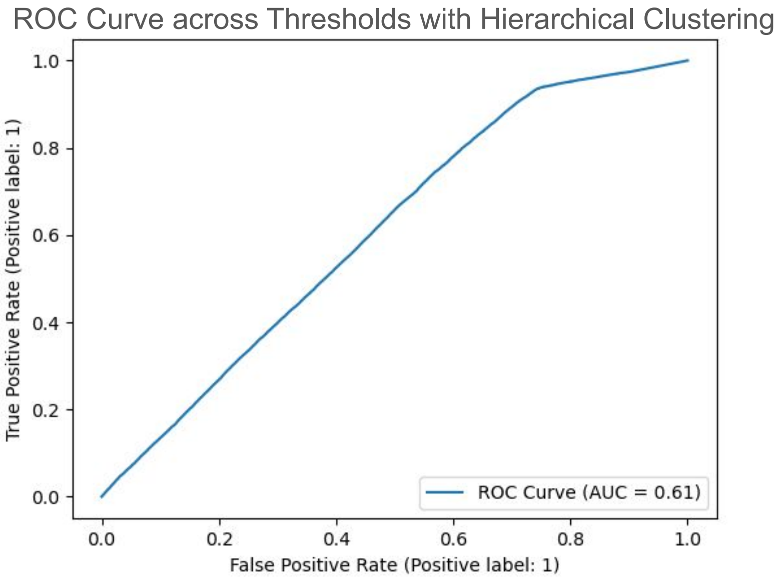
Using one core of Intel I7 processor from 2015 with 16 GB of RAM and no GPU: 3 minutes average time per epoch.

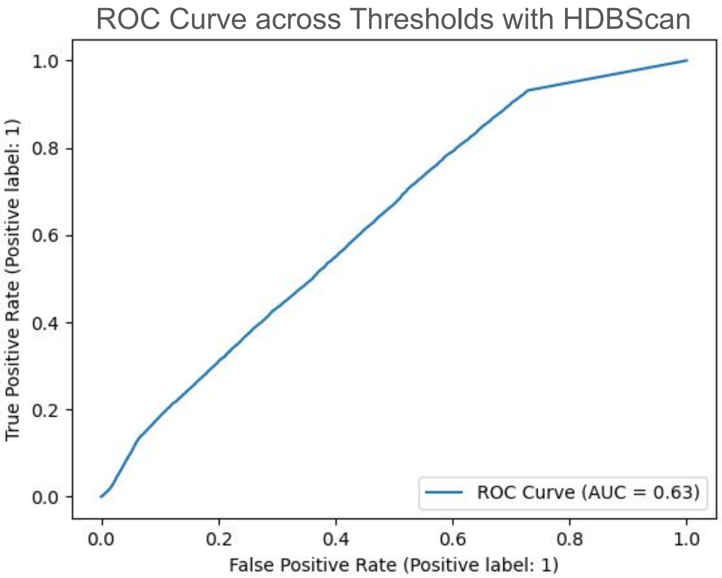
Significantly faster on newer CPUs (< 45 seconds) and likely even faster with CUDA GPU.



Convergence is achieved after ~400 epochs.

Prediction Results on YelpNYC





- Extremely poor results on YelpNYC; Not much better than random guessing.
- It is unclear why this is occurring and requires further investigation.
- Either embeddings are low quality, anomaly metrics are poor or both.

Conclusion

- Pros
  - The only major hyperparameter to set is the anomaly threshold.
  - LightGCN embeddings may be more efficient than other techniques.
  - Explainable and easy to visualize predictions.
- Cons
  - Low quality of predictions when tested on YelpNYC dataset.
  - Requires a lot of memory when using hierarchical clustering.
  - Optimal anomaly threshold hyperparameter varies between datasets.
- Future Work:
  - Test alternative embeddings techniques and loss formulations.
  - Investigate if embeddings or anomaly scores are cause of low performance.
  - Develop improved anomaly metrics that make better use of metadata.
  - Remove need for threshold hyperparameter by looking for abnormally dense groups relative to other similarly sized groups in the dataset.

Thank you!

🙂