Navigating the Dynamics of Financial Embeddings over Time

Antonia Gogoglou [1] **Brian Nguyen** [1] **Alan Salimov** [1] **Jonathan B. Rider** [1] **C. Bayan Bruss** [1]

Abstract

Financial transactions constitute connections between entities and through these connections a large scale heterogeneous weighted graph is formulated. In this labyrinth of interactions that are continuously updated, there exists a variety of similarity-based patterns that can provide insights into the dynamics of the financial system. With the current work, we propose the application of Graph Representation Learning in a scalable dynamic setting as a means of capturing these patterns in a meaningful and robust way. We proceed to perform a rigorous qualitative analysis of the latent trajectories to extract real world insights from the proposed representations and their evolution over time that is to our knowledge the first of its kind in the financial sector. Shifts in the latent space are associated with known economic events and in particular the impact of the recent Covid-19 pandemic to consumer patterns. Capturing such patterns indicates the value added to financial modeling through the incorporation of latent graph representations.

1. Introduction

Credit card transactions represent one form of transferring assets between parties. A graph generated from transactions can be information rich, in that the edges can contain information such as transaction amount and frequency while nodes themselves can contain rich features such as FICO score, income, and account balance. Graphs of financial transactions have some very unique properties not typically studied in traditional social networks. Properties such as extreme power-law distributions and heterogeneity. They also tend to be highly non-stationary in a multi-dimensional way since transactions are processed in a continuous stream and nodes can enter or be removed from each snapshot of this graph. Graph representation learning (GRL) is often used as a generalized approach to feature generation from such a graph structure that can be used in a number of down stream applications. In financial services these include fraud detection and credit decisions (Shumovskaia et al., 2020), (Bruss et al., 2019). Many of the traditional representation learning techniques assume stationarity in the underlying structures that they embed.

To solve the problem of graph representation learning on non-stationary financial graphs, this paper provides a description of our framework for training shallow embeddings from highly dynamic graphs over multiple timeframes that can empower down stream applications. We proceed to perform an extensive in-depth qualitative analysis of embedding shift over time and identification of meaningful shifts. Temporal patterns in the representation space are associated with real world transaction dynamics (e.g. shopping patterns, market changes) and merchant categories with a particular focus on the effects of Covid-19 pandemic to financial transactions. Time series analysis is employed to filter rotational noise from the dynamic retraining process and demonstration of how short-term representation shift can be effectively inferred from prior shift of the embedding space.

2. Related Works

Extending similar efforts from the NLP literature graph approaches to dynamic graph problems have focused on solving the problem of transmitting changing graph information through time while while allowing full expressiveness at each time step. Tracking this temporal information by learning dynamic representation through time using recurrent architectures (Pareja et al., 2019; Goyal et al., 2018) is one approach. Other approaches (Kumar et al., 2019) seek to make explicit the temporal dependencies between graphs by injecting this information into the embeddings themselves by introducing the element of node trajectory. However, the increase of volume of information quickly makes large temporal graph problems intractable in a real world scenario as the number of time steps considered increases. Additionally, the set of time stamps might change dynamically and is not often predefined and/or available from the initialization of the process. Simplifications, such as restricting the change

[*] Equal contribution [1] Capital One. Correspondence to: Antonia Gogoglou <antonia.gogoglou@capitalone.com>.

Proceedings of the 37^{th} International Conference on Machine Learning, Vienna, Austria, PMLR 108, 2020. Copyright 2020 by the author(s).

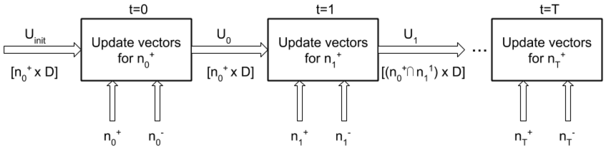

Figure 1. Dynamic training of skip-gram model for merchant embeddings

in size to the number of nodes, are necessary to make up for the increase in graph size as well as increase in parameters for more complicated models.

3. Methods

3.1. Embeddings Generation

The static baseline method for each snapshot of the transaction graph is described in (Bruss et al., 2019). The incoming data are represented in tabular format where each data entry corresponds to a particular transaction between an account and a merchant associated with a given timestamp. A bipartite graph of accounts and merchants connected through transactions can subsequently be inferred from this dataset. We consider each node of a particular type as a *bridge* that facilitates a connection between two nodes of another type by interacting with both of them. In the present work, we focus on connections between *Brand Level Merchants*, meaning that merchants are represented by their brand name without distinguishing between different locations of the same merchant. This approach results in a highly interconnected graph representing spending patterns across the nation.

We form monthly snapshots at the end of each month (in the range 2017-11 to 2020-03) following the aforementioned methodology. One issue that has been contemplated in literature (Levy & Goldberg, 2014; Hamilton et al., 2017) is the random rotation of the embedding space in the skip-gram model. This rotation would constitute each snapshot's embeddings radically different in values compared to the previous and therefore their use in downstream models would be compromised. To address this matter, we opt for a warm-start training where each snapshot's model is initialized with the previous month's final state. Once a merchant has entered the embedding space its position is maintained and receives updates only when new training pairs appear that include this merchant. Considering a set of T snapshots which correspond to time stamps $t_0 < t_1 < ... < t_T$. For each time step t_i, n_i^+ positive context pairs are generated based on co-occurrence of transactions within a given time window, while n_i^- pairs are sampled for negative context. Details of training process are depicted in Figure 1.

3.2. Quantifying Shift

In order to associate *representation shift* as measured by first-order statistics in the latent space with real-world shift, i.e. *semantic shift*. Our selected metrics are *Euclidean distance* and *cosine distance* to measure both magnitude shift by the absolute values of the embedding dimensions but also similarity shifts as measured by the angle between embedding vectors. For a time stamp t with $t \in [0, T]$, $\Delta_{mag,t}$ and $\Delta_{cos,t}$ are defined as:

$$\Delta_{mag,t} = ||v_t - v_{t-1}||_2 \tag{1}$$

$$\Delta_{cos,t} = 1 - \frac{v_t \cdot v_{t-1}}{||v_t|| \cdot ||v_{t-1}||} \tag{2}$$

Of particular interest is the timing of maximum shift for each merchant or a subset of them. We add normalization by the cumulative shift of all the nodes N within time stamp t to ensure that the shift ranking is sensitive to the nodes that drifted more compared to the drift of the rest of the nodes. For each merchant across a set of time stamps T, the maximum shift is defined as:

$$max\Delta_{mag,t} = argmax_t\left(\frac{\Delta_{mag,t}}{\sum^N \Delta_{mag,t}}\right) \tag{3}$$

3.3. Time Series Analysis

Shift trajectories In order to formalize the quantification of shift over time we aggregate the first order statistics across different timeframes that differ by δt and create a time series of shifts:

$$\tau_{\Delta_{cos,t}} = \{\Delta_{cos,t_0}, \Delta_{cos,t_0+\delta_t}, ..., \Delta_{cos,T}\} \tag{4}$$

The resulting time series may be viewed as a *state space* model where individual components such as a financial trend component or a seasonal one are combined to produce the observed values of shift. By decomposing the time series into its constituent components we aim to identify the effects of the trend component as well as the expected random component that arises from the rotation of the embedding space after multiple rounds of training. A particular method of time series filtering that has been utilized in conjunction with embedding approaches is *Kalman filtering* (Kumar et al., 2019; Bamler & Mandt, 2017). The Kalman filter operates on a series of measurements observed over time, e.g. the embedding shift, and assumes that they contain some level of Gaussian noise or inaccuracy, using the time series to estimate $P(x_t|z_{0:T-1})$ as given by the equations below.

$$x_{t+1} = A_{txt} + b_t + \mathcal{N}(0, Q_t) \tag{5}$$

$$z_t = C_t x_t + d_t + \mathcal{N}(0, R_t) \tag{6}$$

We applied Kalman smoothing to embeddings (normalized using $\Delta_{mag,t}$) elementwise, by assuming an independent difference vector, namely *velocity*, for each embedding component and calculating the next time step components as a linear combination of predicted and actual values calculated by an EM algorithm. We therefore compute the normalized difference vector $\Delta\hat{v}_t \in \mathcal{R}^d$ or *velocity*, where d is the dimension of the embedding vectors.

$$\Delta\hat{v}_t = \hat{v}_t - \hat{v}_{t-1} \tag{7}$$

Finally, a predictive sequence model is applied to attempt inference of future shifts given past observations. If there are meaningful short-term and/or long-term dependencies then a sequence model should be able to identify them and outperform a naive average based baseline. A Long Short Term Memory (LSTM) model for regression is utilized to predict shift in Δ_t time stamps in the future by looking back at l time stamps.

4. Results

Semantic Shift The question arises whether all merchant embeddings shift at the same pace or whether there exist categories that showcase larger shifts in particular time periods. We measure the percentage of each merchant category and repeat this calculation amongst the top 10,000 shifting merchants (according to $\Delta_{cos,t}$) between different time stamps. In Figure 2 we report results for time periods of interest. The first chart showcases the effect of seasonal spending patterns such as holiday seasons that cause the categories of Retail and Social to exhibit the largest shift between December and February. Services as well as Travel follow a similar trend. An interesting pattern appears in the next two charts, by comparing the shift in 2019-03 and 2020-03. It appears that Social, Retail and Services achieve higher percentages amongst the maximum shifting merchants compared to the same time last year, which is in line with the effects of the Covid-19 pandemic in consumer behavior. Similar patterns appear when we explore the maximum shift in magnitude $max\Delta_{mag,t}$ from Equation 3. From the time series of $\Delta_{mag,t}$ for each merchant, the month in which the merchant exhibited the highest shift compared to the total magnitude shift of all merchants in that same month is selected as *max shift month*. By displaying the counts of merchants that exhibit their max shift in each given month, we notice trends appearing that coincide with known financial events. In the beginning of 2019 and then again in the summer of 2019 market volatility may have influenced consumer behavior. Interestingly, counts appear to spike in the first few months of 2020 with a peak in March 2020 when the Covid-19 pandemic caused major changes in spending patterns.

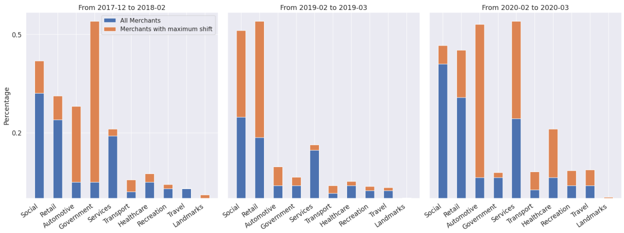

Figure 2. Percentages of different merchant categories in the set of most frequent merchants and in the set of top 10,000 shifting merchants

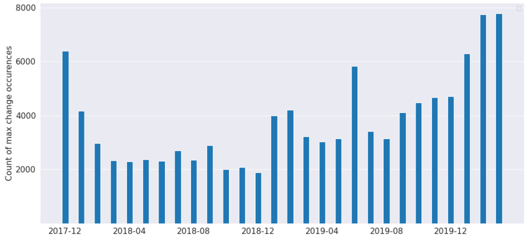

Figure 3. Normalized count of merchants that exhibit their maximum shift in each time stamp.

Visualizing and filtering trajectories We proceed to explore the sequence of $\Delta_{cos,t}$ as a whole and attempt to identify its constituent components. We calculate 120 different 2 hop neighborhoods with $k = 100$ neighboring nodes and, after normalizing these embeddings using $\Delta_{magn,t}$, we smooth them using Kalman smoothing on a per-merchant basis and take their element-wise difference per time stamp to generate velocity vectors. Subsequently, the velocity vectors $\Delta\hat{v}_t$, from equation 7, of all merchants across all time stamps are aggregated and passed through t-SNE dimensionality reduction. As we observe in Figure 4, for the non smoothed velocity embeddings two clusters emerge that correspond to regions of high and low frequency. After smoothing (second chart), the frequency pattern was maintained and by shading according to $\Delta_{cos,t}$ quartiles we notice higher degree of separation among the clusters high frequency region. In these clusters, the denser ones appear to be correlated with high $\Delta_{cos,t}$. In the last chart of Figure 4 we observe that neighborhood memberships expand across regions of zero and non-zero movement (low and high frequency) which indicates that the inclusion of non-updating nodes in our sequential training workflow allows nodes to maintain connections from previous time frames and expand their neighborhood in both highly changing and more stable regions.

Finally, we explore the predictability of representation shift as measured by $\Delta_{cos,t}$ and attempt to identify the effect

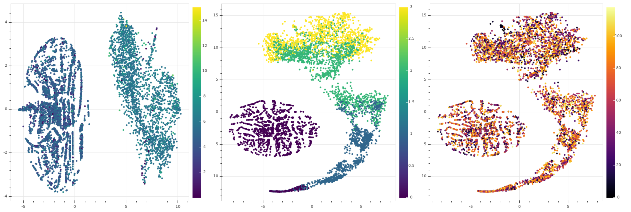

Figure 4. t-SNE of 2020-03 velocity vectors shaded by log frequency, quartile cosine distance, and neighborhood membership out of 120 neighborhoods. Center and right are Kalman smoothed, while left chart is raw embeddings.

of long- and short-term dependencies in the latent space. Sequence and training length for our experiments varies between 1 and 7 months, meaning $\Delta_{cos,t}$ of each month with the previous one are available for 1 to 7 months. A naive baseline is added as a frame of reference, where we estimate the next month's $\Delta_{cos,t}$ as a moving average of the past sequence values. For performance evaluation Mean Square Error is reported. Results are shown in Tables 1 and 2 for the test month of 2 different time periods. Steady increase in performance is observed over the baseline for all sequence and training lengths which indicates that non trivial temporal and perhaps repetitive patterns arise in the embedding space. It appears that extending the sequence length up to the 5 previous time steps achieves the highest performance; however increasing the sequence length offers limited to no benefit. This could be attributed to seasonal trends that move embeddings further than their representation 7 time steps ago as well as the overall rotation of the embedding space. Moreover, error values are on average higher for March 2020 when an unprecedented change in consumer behavior occurred due to the Covid-19 pandemic.

Table 1. Mean Square Error ($\times 10^4$) for naive baseline model and LSTM with training month up to 2019-12 and test month 2020-01

sequence length	training length				
	Baseline	LSTM			
		1	3	5	7
1	1.75	1.08	0.95	0.94	0.92
3	1.52	0.68	0.64	0.62	0.61
5	1.37	0.62	0.56	0.56	**0.47**
7	1.38	0.63	0.59	0.58	0.50

5. Conclusions

We introduce a dynamic scalable graph representation workflow deployed on financial graphs and a qualitative analysis is performed to extract patterns from the evolution of the consumer behavior over time.

Table 2. Mean Square Error ($\times 10^4$) for naive baseline model and LSTM with training month up to 2020-02 and test month 2020-03

sequence length	training length				
	Baseline	LSTM			
		1	3	5	7
1	1.38	0.95	0.91	0.91	0.89
3	1.30	0.70	0.69	0.70	0.69
5	1.21	0.69	0.68	0.68	**0.62**
7	1.26	0.70	0.69	0.68	0.63

References

Bamler, R. and Mandt, S. Dynamic word embeddings. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 380–389. JMLR. org, 2017.

Bruss, C. B., Khazane, A., Rider, J., Serpe, R. T., Gogoglou, A., and Hines, K. Deeptrax: Embedding graphs of financial transactions. *2019 18th IEEE International Conference On Machine Learning And Applications (ICMLA)*, pp. 126–133, 2019.

Goyal, P., Chhetri, S. R., and Canedo, A. dyngraph2vec: Capturing network dynamics using dynamic graph representation learning. *CoRR*, abs/1809.02657, 2018. URL http://arxiv.org/abs/1809.02657.

Hamilton, W. L., Ying, R., and Leskovec, J. Inductive representation learning on large graphs. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, pp. 1025–1035, Red Hook, NY, USA, 2017. Curran Associates Inc. ISBN 9781510860964.

Kumar, S., Zhang, X., and Leskovec, J. Predicting dynamic embedding trajectory in temporal interaction networks. In *Proceedings of the 25th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2019.

Levy, O. and Goldberg, Y. Neural word embedding as implicit matrix factorization. In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'14, pp. 2177–2185, Cambridge, MA, USA, 2014. MIT Press.

Pareja, A., Domeniconi, G., Chen, J., Ma, T., Suzumura, T., Kanezashi, H., Kaler, T., and Leiserson, C. E. Evolvegcn: Evolving graph convolutional networks for dynamic graphs. *CoRR*, abs/1902.10191, 2019. URL http://arxiv.org/abs/1902.10191.

Shumovskaia, V., Fedyanin, K., Sukharev, I., Berestnev, D., and Panov, M. Linking bank clients using graph neural networks powered by rich transactional data. 2020.