PayVAE: A Generative Model for Financial Transactions

Niccolò Dalmasso[1,2] **and Robert E. Tillman**[2] **and Prashant Reddy**[2] **and Manuela Veloso**[2]
[1]Department of Statistics & Data Science, Carnegie Mellon University
[2]J. P. Morgan AI Research, New York

Abstract

Financial transactions, such as payments between individuals and institutions, are ubiquitous data both in large financial institutions and in peer-to-peer payment services and online marketplaces. Such data have a rich latent structure consisting of both a temporal component, the point process of times at which transactions occur, and a relational component, the patterns of payments between different entities. Most existing approaches either model individual point processes independently, without regard to the relational component, or instead focus on the evolution of the relational structure and ignore the point process of transaction times. We thus propose `PayVAE`, a generative model which attempts to learn the temporal and relational structure of financial transactions directly from the data. We apply our model to a real peer-to-peer payments dataset and demonstrate that it is capable of generating realistic transactions.

Introduction

Financial transactions, such as payments between individuals and institutions, are ubiquitous data in large financial institutions, peer-to-peer payment services and online marketplaces. Wholesale payments businesses, which provide clients with solutions to facilitate payments to and from other entities, represent large and stable sources of revenue for banking institutions (Allchin et al. 2018). Electronic funds transfer networks such as ACH, SWIFT, Fedwire and Giro facilitate billions of annual transactions between institutions, governments, businesses and individuals (Gerdes et al. 2020). Similarly, peer-to-peer payments businesses such as PayPal, Venmo, Square and Zelle, which facilitate everything from friends splitting restaurant bills to paying rent and for goods and services, process increasing volumes of transactions each year (Rotatori 2018).

We define *transaction data* as sequence data comprised of a *sender*, *receiver* and *transaction time*. Other variables such as a payment amount or type may also be included as well as variables that are specific to the sender or receiver but do not vary between transactions. We provide (synthetic) examples of transaction data from a wholesale payments business (Borrajo, Veloso, and Shah 2020) and a peer-to-peer payments business (Venmo 2020) in Figure 1.

Such data have a rich latent structure consisting of both a *temporal* component and *relational* component. Transaction times form a *temporal point process*, or a random process realized as a sequence of times an event occurs. Temporal point processes directly model periods between events as random variables and do not require choosing a time window to aggregate events (Daley and Vere-Jones 2003). This makes them a natural choice for modelling the times of irregularly occurring transactions. They have previously been used in finance to model arrival times of buy and sell orders on exchanges and other activities (Bacry, Mastromatteo, and Muzy 2015). Aside from the temporal structure, the patterns of interactions between senders and receivers form a rich relational structure. Most entities are likely to transact with a relatively small set of other entities. Transitive relations between these sets may form communities from which complex interactions may be observed. Such relational structure has been modeled extensively using networks or graphs.

Parametric models (Hawkes 1971; Isham and Westcott 1979; Aalen, Borgan, and Gjessing 2008) and more recently Generative Adversarial Networks (GANs) (Xiao et al. 2018) have been proposed as generative models for temporal point process data. However, these approaches only consider individual point processes independently; they do not consider the complex relational structure in point processes involving multiple entities. We are not aware of existing generative models capable of simultaneously learning the temporal and relational structure of such transactions directly from data.

In this paper, we propose `PayVAE`, the first generative model for transaction data that attempts to learn both the point process of transaction times as well as the complex relations between senders and receivers directly from the data. Our proposed model is based on a variational autoencoder (VAE) architecture where we apply convolutions to transaction times across interacting senders and receivers and use a recurrent model to encode the sequential structure. We demonstrate that our model generates realistic transactions by applying it to a real peer-to-peer payments dataset and comparing the real and simulated data.

Advances in popular neural generative architectures such as GANs and VAEs have led to increased interest in generative models. In finance, there are specific motivations for working on generative models. Due to the highly regulated nature of the industry, sharing data and research collabora-

```
Time_step            2020-10-12 16:19:25
Transaction_Id       QUICK-PAYMENT-57485
Sender_Id                   CLIENT-57463
Sender_Country                       USA
Sector                                18
LOB                                  CCB
Bene_Id                     CLIENT-57464
Bene_Country                    BARBADOS
USD_Amount                       100.197
Label                                  0
Transaction_Type           QUICK-PAYMENT
```

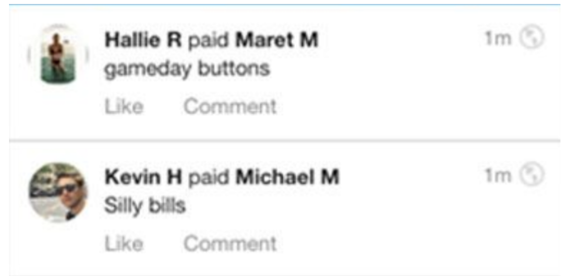

Figure 1: Synthetic transactions from a wholesale payments business (left) and peer-to-peer payments business (right)

tions are often complicated both with external researchers as well as between internal groups at the same institution. Generative models capable of producing high-fidelity synthetic data provide an alternative means for sharing data and collaborating when real data are highly confidential (Assefa et al. 2020). Additionally, a number of problems in finance, e.g. fraud detection, involve extreme class imbalances. Generative models can be co-opted to improve the robustness of systems trained on such data by producing more diverse examples of the imbalanced class for training.

Background and Related Work

Temporal Point Processes

Temporal point processes have been successfully applied across many domains including medical records (Weiss and Page 2013; Lian et al. 2015), social media analysis (Farajtabar et al. 2014, 2016b; Zipkin et al. 2016) and finance (Bacry, Mastromatteo, and Muzy 2015). Temporal point processes model the *conditional intensity function*, i.e. the rate at which the next event is expected to occur given past events. The literature on temporal point processes is vast and has been driven by adapting the conditional intensity function to capture different types of events. Examples of parametric models include self-exciting processes or Hawkes processes (Hawkes 1971), self-correcting point processes (Isham and Westcott 1979) and survival processes (Aalen, Borgan, and Gjessing 2008). Recent work has used non-parametric models such recurrent neural networks to estimate the conditional intensity function directly (Du et al. 2016; Xiao et al. 2017). This work has been extended to multivariate point processes (Mei and Eisner 2017) where different events types are modeled simultaneously; however, no relational structure is included in the modeling step apart from pairwise correlations between event types. More recently, Xiao et al. (2018) proposed a GAN-based approach for direct sequence emulation, but this approach is only applicable to one-dimensional event sequences. *Spatio-temporal point processes* extend temporal point processes to account for spatial locations of events; see González et al. (2016) and Reinhart (2018) for recent reviews. In general, the occurrence of an event at a specific location and time is modeled based on past events within a neighborhood of such spatial location. This is usually achieved via Gaussian process regression or parametric models of the conditional intensity function (Cressie and Wikle 2011; Diggle 2014). Recent work has extended spatio-temporal point processes to graphs (Ioannidis, Romero, and Giannakis 2018), however assuming the underlying network exists and is available during the training phase.

Dynamic Graph Embeddings

Graph representation learning is an extremely active research area. While most approaches focus on static graphs, there are a number of recent approaches which model the evolution of dynamic graphs over time. Most treat the temporal component as a regular discrete interval – they do not model a point process in continuous times. Popular methods based on matrix factorization (Li et al. 2017), random walks (Nguyen et al. 2018; Yu, Yin, and Zhu 2018) and combinations of Graph Neural Network (GNN) and recurrent architectures (Seo et al. 2018; Yu, Yin, and Zhu 2018; Goyal et al. 2017, 2020; Pareja et al. 2020) have been proposed.

In contrast, Trivedi, Dai, and Song (2017); Trivedi et al. (2019); Zuo et al. (2018); Farajtabar et al. (2016a); Chang et al. (2020) propose embeddings based on the continuous arrival and removal times of nodes and links, e.g. as in a social network, and/or continuous time processes evolving over the graph structure. A key distinction, however, is these approaches assume the existence of an observable graph with persistent links at any given time, e.g. current friend/follower relationships between users in a social network. In our setting, we only observe instantaneous transactions between senders and receivers; there are no persistent links. While we assume these transactions are influenced by rich latent relational dynamics that change over time, we are not able to directly observe these relationships as a graph.

More fundamentally, these approaches are designed to solve specific or general downstream prediction tasks; to our knowledge there are none which are generative models capable of producing synthetic transaction data with high fidelity and diversity. A further distinction between our approach and these methods is they use intensity functions to model point processes whereas `PayVAE` is intensity-free.

Variational Autoencoders

A *variational autoencoder* (VAE, Kingma and Welling (2013)) is a generative model based on a directed Bayesian graphical model in which the observable data $\mathbf{x}$ are assumed to be generated from a set of latent variables $\mathbf{z}$. Starting from a simple prior $p_\theta(\mathbf{z})$ (usually multivariate Gaussian), a

VAE is composed of two components, both usually approximated using neural networks: a decoder which learns the likelihood $p_\theta(\mathbf{x}|\mathbf{z})$ and an encoder which approximates the intractable posterior of the latent space $p_\theta(\mathbf{z}|\mathbf{x})$ with a distribution $q_\phi(\mathbf{z}|\mathbf{x})$. The VAE aims to maximize the marginal likelihood of the resulting observed data $p_\theta(\mathbf{x})$. However, as the term is intractable on its own, the evidence lower bound (ELBO) is maximized instead:

$$\log p_\theta(\mathbf{x}) \geq \mathbb{E}_{\mathbf{z}\sim q_\phi(\mathbf{z}|\mathbf{x})}\left(\log p_\theta(\mathbf{x}|\mathbf{z})\right) - \mathrm{D}_{\mathrm{KL}}\left(q_\phi(\mathbf{z}|\mathbf{x})||p_\theta(\mathbf{z})\right),$$

where D_{KL} is the KL-divergence between the posterior approximation and the prior. As a note, the above bound becomes an equality if the posterior is approximated exactly, i.e. $q_\phi(\mathbf{z}|\mathbf{x}) = p_\theta(\mathbf{z}|\mathbf{x})$. While the VAE architecture is structurally similar to the simple *autoencoder* (AE) architecture, the key distinction is that in an AE, $\mathbf{z}$ is a vector, whereas in a VAE, $\mathbf{z}$ is sampled from a distribution. AEs are not suitable for generative modelling because the learnt latent space of $\mathbf{z}$ may be discontinuous, producing unrealistic outputs for some values and reconstructing the original training data rather producing diverse variations. VAEs avoid this problem by explicitly defining $\mathbf{z}$ to be continuous.

Generative modelling using VAEs is a very active research area with many successful applications; see Kingma and Welling (2019) for an extensive review. Related to our work, variational graph autoencoders (Kipf and Welling 2016) allow for the generation of graph-like structures, provided the underlying relational structure is known and available during training. VAEs have also been used for generating samples of stochastic point processes of sets of discrete actions (Mehrasa et al. 2019), which are modelled simultaneously without including any relational structure. Finally, a related application is frame-to-frame video prediction (Denton and Fergus 2018; He et al. 2018; Franceschi et al. 2020), which relies on smooth changes between subsequent frames.

Proposed Model

There are two main challenges to building a generative model for financial transactions: first, the model should simultaneously learn both the relational and temporal structure from the data without assuming the availability of an observable network of relations during training; second, the architecture should scale to a large number of accounts C, as current methods in the literature would require the creation of $\mathcal{O}\left(2C^2\right)$ event types to account for all possible transactions between accounts. Our proposed model addresses both of these challenges, as highlighted by the example in the following evaluation section where we successfully generates realistic synthetic samples for transactions involving 625 accounts.

Data Preparation

As mentioned in the introduction, in this work we assume transaction data are sequences of a sender, receiver and transaction time, excluding other additional variables. Let $t^k_{i,j}$ be the time of the k^{th} transaction between account $i, j \in \{1, ..., C\}$ such that account i is the sender and account j is the receiver. As noted in Xiao et al. (2018), using inter-arrival times rather than transaction times makes training easier and more robust. We define the inter-arrival time $\tau^k_{i,j}$ as the time elapsed between two consecutive transactions, i.e. $\tau^k_{i,j} = t^k_{i,j} - t^{k-1}_{i,j}$. We assume that $\tau^0_{i,j} = 0$ for all accounts i and j. In addition, if there are only k such transactions between accounts i and j in a given sequence, we set $\tau^{k+1}_{i,j}, \tau^{k+2}_{i,j}, ..., \tau^{k^\star}_{i,j} = 0$, where $k^\star$ is the maximum number of transactions between any two accounts in the training data. In this work, a transaction sequence corresponds to all transactions occurred in a day.

Given the above setup, we want to transform the transaction sequence in a format that makes training the model more efficient while simultaneously avoiding information loss; that is, we want to be to able recover the original transaction data regardless of the training formatting process. To this end, we capture the relational structure between accounts using a matrix $A \in \mathbb{R}^{C\times C}$ in which each entry is equal to the inter-arrival time of a transaction, i.e. $A_{i,j} = \tau_{i,j}$. We create such inter-arrival times matrices $\{A^k\}^{k^\star}_{k=1}$ for each of the k^{th} transactions, where $k^\star$ is the maximum number of transactions between any two accounts across a sequence in the training data. We combine all A^k matrices into an *inter-arrival times* (IAT) *tensor* $T \in \mathbb{R}^{C,C,k^\star}_+$, in which the entry $T^k_{i,j}$ records $\tau^k_{i,j}$, the inter-arrival time of the k^{th} transaction between the sender i and the receiver j (or 0 if no such transaction occurs). We also define a binarized version of the IAT tensor as Y^k, where $y^k_{i,j} = \mathbb{I}\left(A^k_{i,j} > 0\right)$.

Encoding Accounts Relational Structure

Capturing each accounts' relational dynamics from the IAT tensor is challenging for a neural network layer, as operations such as graph convolutions (as in `GraphVAE` (Simonovsky and Komodakis 2018)) are not applicable. Figure 2, left, highlights were the relevant information for the first account are encoded in the IAT tensor in its current form: the outgoing inter-arrival times are available on the first row while the incoming inter-arrival times are on the first column. In addition, changing the position of one account in the row/column changes the encoded information, i.e. the ordering of accounts might affect the model's predictive capacity even though there is no natural known ordering of accounts. We propose a novel transformation to address these issues, shown in Figure 2, right. The matrix $A \in \mathbb{R}^{C\times C}$ is transformed into a wider matrix $\tilde{A} \in \mathbb{R}^{C\times(2C-1)}$, in which the inter-arrival times for each account's incoming transactions are appended at the end of each row. Duplicating the information allows us to capture all the relevant information for a specific account with a convolutional layer using a horizontal kernel. This transformation is lossless and the information encoded can be captured by a convolutional layer regardless of the account ordering.

`PayVAE` Architecture

Figure 3 provides a schematic representation of our proposed architecture, `PayVAE`. The encoder is composed of

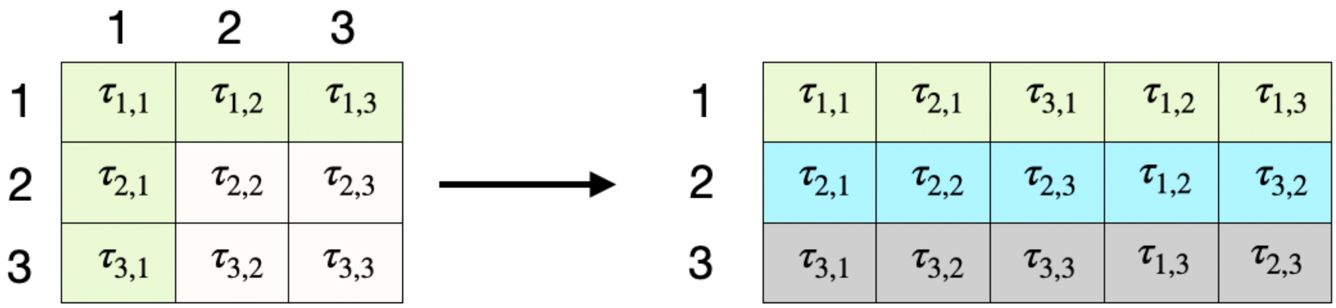

Figure 2: A simple example with 3 accounts depicting how the proposed transformation modifies the inter-arrival time matrix. By appending the outgoing transaction inter-arrival times as new columns to each account, we create a representation which can be easily captured by a convolutional layer with a horizontal kernel.

two layers, a convolutional layer with a horizontal kernel and a recurrent neural network layer. Specifically, we first pass the transformed IAT tensor through the convolutional layer with a horizontal kernel of size $2C-1$, with no striding or padding (Goodfellow, Bengio, and Courville 2016, Chapter 9); in other words, the inter-arrival times for each row in the transformed IAT tensor (i.e., for each account) are convolved together, with the convolution being applied row-wise. This operation results in a C-dimensional intermediate representation, which is then fed as an input to an LSTM (Hochreiter and Schmidhuber 1997) to capture the temporal structure. We map the output of the encoder into a d-dimensional latent space, where d is a model hyper-parameter. The decoder was built mirroring the encoding process. After we sample from the latent space, we feed the sample as input to a recurrent layer (again an LSTM in this case) to build a C-dimensional representation of the sample. We split the reconstructed output in two parts, where we separately simulate the binarized version of the inter-arrival times Y and the inter-arrival times A. We found reconstructing both tensors provided better results than simply reconstructing inter-arrival times alone. Both reconstruction are obtained by applying a transposed convolution layer (Dumoulin and Visin 2016). Although technically not exactly representing an inverse convolution operation, a transposed convolution layer allows the intermediate representation to be "up-sampled" and greatly improves the model scalability by reducing the output size of the recurrent layer by an order of magnitude, from $\mathcal{O}(2C^2)$ to $\mathcal{O}(C)$ respectively.

Loss Function

Let $\tilde{y}^k_{i,j}$ and $\tilde{A}^k_{i,j}$ being the reconstructed probabilities and inter-arrival times for the binarized and actual IAT tensor respectively. In order to take into account both output tensors, we consider a mixed bernoulli-Gaussian model, for which:

$$\log\left(p(Y, A|\mathbf{z})\right) \propto \sum_{i,j,k} y^k_{i,j} \log\left(\tilde{y}^k_{i,j}\right) + (1 - y^k_{i,j}) \log(1 - \tilde{y}^k_{i,j}) + \log\left(\mathcal{N}(A^k_{i,j} | \tilde{A}^k_{i,j}, \sigma^2)\right),$$

which in practice is equivalent to a binary cross-entropy term for the reconstructed probability and a squared-loss term for the inter-arrival times. During our experiments, we set $\sigma = 1$ and found no significant differences when restricting the latter term only to non-zero inter-arrival transaction times (which makes loss calculations more efficient).

Evaluation

We evaluate `PayVAE` by applying it to a real transaction dataset and comparing the the simulated transactions to the real data, taking fidelity and diversity into consideration.

Dataset

We use a publicly available dataset consisting of over 7 million peer-to-peer payments scraped from the Venmo public API between July 2018 and February 2019 (Salmon 2019). Because many of the accounts in this dataset transact infrequently during this time period, we selected a subset of accounts which regularly used Venmo as a payment method for poker games. We focus on this subset due to their high levels of activity and interesting patterns of interactions. After filtering transactions to include only those whose descriptions include the term "poker", we selected the top 10 accounts which transacted with the most other accounts, i.e. the high degree nodes. We select the subset of 625 accounts which are separated from any of these 10 accounts by up to 5 degrees via payments made during this time period. The resulting dataset, which includes only these 625 accounts, consists of 1,574 transactions occuring over 40 days where between 17 and 71 transactions occur on a given day.

Experimental Setup

We treat each day as a sample in our training data and construct IAT tensors for each. We train for 100,000 epochs with early stopping when the median loss over the last 1,000 epochs exceeds the previous 1,000 using an NVIDIA T4 GPU. We set the dimension d of the latent space to 1024. We use the `Adam` optimizer with learning rate $1e^{-4}$. We assume both VAE priors and approximate posteriors to be Gaussian, hence deploying the loss function in Kingma and Welling (2013, Section F). After training, we generate synthetic transactions by sampling from the latent space and using the decoder of `PayVAE` to generate synthetic IAP and IAT tensors. We simulate an equal number of days as

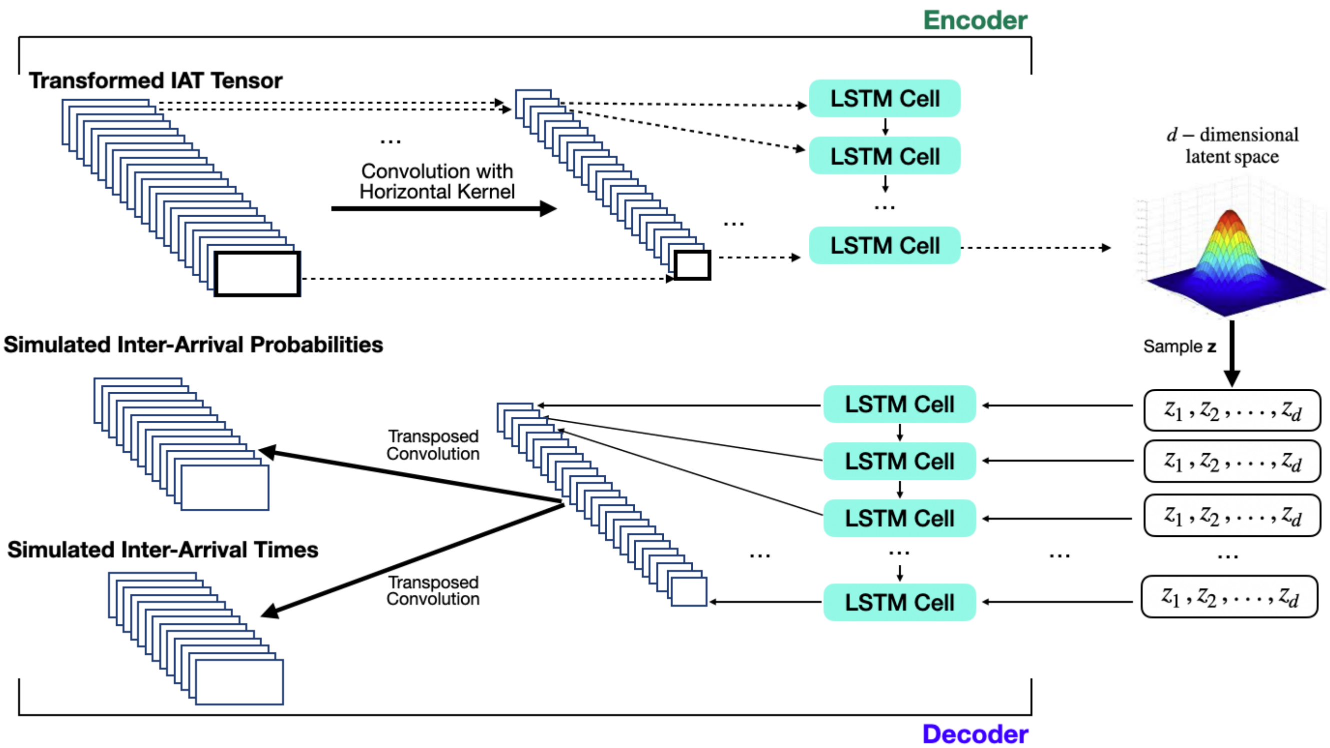

Figure 3: Schematic of the PayVAE architecture. The encoder is composed of horizontal convolutional and recurrent (LSTM) layers to capture both relational and temporal structure. Samples from the d-dimensional latent space are decoded by first passing through a recurrent layer (LSTM), and then up-sampled by a transposed convolutional layer to generate both simulated inter-arrival probabilities and times.

present in the training set (40). We then select a threshold such that the number of entries in the IAP tensor exceeding this threshold is roughly equivalent to the number of transactions in the real dataset. The resulting synthetic dataset consists of the transactions represented in the IAT tensor which correspond to entries in the IAP tensor that exceed this threshold.

Evaluation Metrics

To assess temporal and relational fidelity and diversity, we compare the distributions of transaction times and occurrences of transactions across days and accounts. We first consider the *transaction times* of the first, second and third transactions between any two parties on a given day in the real and simulated data; Figure 4, top row, compares the histograms for each. We also compare the histograms of the *numbers of transactions* that occur in a day for the real and simulated data in Figure 4, bottom row. Finally, we construct graphs for the real and synthetic data such that nodes correspond to accounts and directed edges indicate whether a payment occurs between a sender and receiver. We do this both on a per day basis and also across the entire 40 days. In Figure 4, bottom row, we provide histograms of the *daily account degrees*, or numbers of outgoing and incoming transactions for an account on a given day as well as the *overall account degrees*, or number of outgoing and incoming transactions for an account over the entire 40 day period.

Analysis

We observe that the distributions of synthetic transaction times and numbers of transactions per day closely resemble the real data, indicating both strong fidelity within a sample and diversity of transaction times. The distribution of first transaction times most closely resembles the real data, capturing the bimodal nature of the distribution. The similarity decreases for each successive transaction, as the real data become more sparse. When we consider the degree distributions, however, we notice that while within a given sample the degree distributions of the real and simulated data closely match, i.e. most accounts have no transactions on a given day with a small number transacting once and very few transacting more than once, the overall degree distributions do not match very well. Upon inspection, this appears to be due to limited diversity in the number of accounts that are active across samples in the simulated data, i.e. each account that is active, or involved in a transaction, tends to interacts with a smaller number of accounts across the entire dataset than in the real data. In summary, while the fidelity is strong for each sample generated by PayVAE and the transaction times reflect the diversity of the real data, the pairs of accounts involved in a transaction tend to repeat more often across samples than in the real data, indicating limited diversity with respect to active accounts in the simulated data.

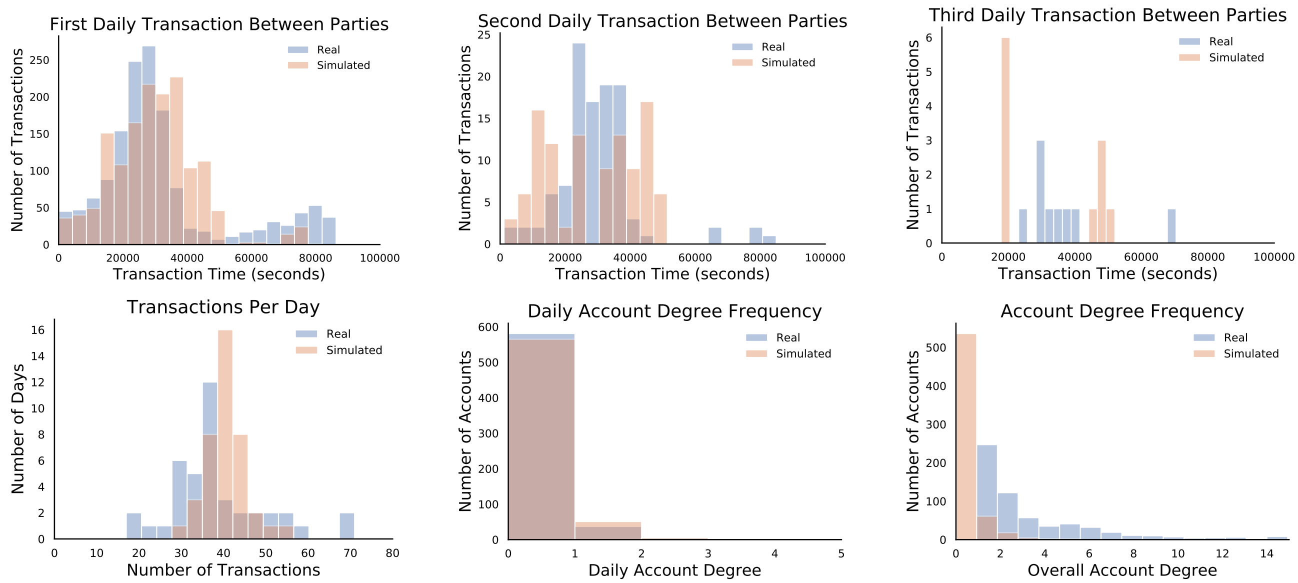

Figure 4: Distributions of evaluation metrics for 40 real and synthetic transaction sequences from the Venmo data. *Top Row*: Time of first, second and third daily transactions between accounts. *Bottom Row*: Number of transactions per day and daily and overall account degree frequency. For the definitions of these metrics see text.

Conclusion and Future Work

We proposed PayVAE, a variational autoencoder which attempts to model both relational and temporal structure of financial transaction data. Our model relies on a novel scheme for transforming transaction sequences into a tensor which promotes efficient training and is invariant to the ordering of accounts (Figure 2). The encoding architecture is composed of a convolution layer equipped with a horizontal kernel, followed by a recurrent neural network. The decoding architecture mimics the encoder, relying on transposed convolution to scale simulations to a large number of accounts. We demonstrated the efficacy of PayVAE by applying it to a public peer-to-peer payments dataset consisting of transactions between 625 accounts over 40 days; we found it is able to simulate realistic daily sets of transactions which closely mimic the transaction times and occurrences of transactions in the real data, but tends to generate samples with limited diversity in terms of the accounts which are active. To the best of our knowledge, PayVAE is the first generative model for learning the temporal and relational structure of transaction data and the first such model that does not rely on directly modeling the point process intensity function.

In future work, we aim to improve the diversity in the number of active accounts generated by PayVAE, potentially by improving the architecture or encoding scheme or devising an appropriate loss penalty. We also plan to include additional transaction variables, such as transaction amount and payment type and investigate the scalability of the architecture to very large numbers of accounts, e.g. wholesale payments businesses process payments for accounts in the order of millions. Specifically, we are considering developing differentiable sparse layers that are compatable with our architecture for efficiency (current modules which support sparse tensors are extremely limited in most popular autograd systems). Additionally, in the present version of PayVAE the entire transaction sequence is mapped into a single d-dimensional latent space; another approach is to represent the temporal structure in the latent space as in temporal difference VAEs (TD-VAE, Gregor et al. (2019)). We also note that while PayVAE is a generative model, the representation learned could be utilized for downstream tasks such as anomaly detection (for which VAEs have shown promising performance (Xu et al. 2018)); this could be relevant for detecting fraudulent financial activities. Finally, we are currently working with a large whole payments business to apply PayVAE to a large wholesale payments network.

Acknowledgments

The authors are grateful to the two reviewers for their insightful comments which improved the quality of the paper.

Disclaimer

This paper was prepared for informational purposes by the Artificial Intelligence Research group of JPMorgan Chase & Co and its affiliates (“J.P. Morgan”), and is not a product of the Research Department of J.P. Morgan. J.P. Morgan makes no representation and warranty whatsoever and disclaims all liability, for the completeness, accuracy or reliability of the information contained herein. This document is not intended as investment research or investment advice, or a recommendation, offer or solicitation for the purchase or sale of any security, financial instrument, financial product or service, or to be used in any way for evaluating the merits of participating in any transaction, and shall not constitute a solicitation under any jurisdiction or to any person, if such solicitation under such jurisdiction or to such person would be unlawful.

References

Aalen, O.; Borgan, Ø.; and Gjessing, H. 2008. *Survival and Event History Analysis: A Process Point of View*. ISBN 0-387-20287-0. doi:10.1007/978-0-387-68560-1.

Allchin, C.; Davis, J.; O'Kelly, R.; Rosberg, M.; and Sommer, M. 2018. Wholesale Payments: Disrupt from Within. Technical report, Oliver Wyman.

Assefa, S. A.; Dervovic, D.; Mahfouz, M.; Tillman, R. E.; Reddy, P.; Balch, T.; and Veloso, M. 2020. Generating synthetic data in finance: opportunities, challenges and pitfalls. In *Proceedings of the 1st ACM International Conference on AI in Finance*.

Bacry, E.; Mastromatteo, I.; and Muzy, J.-F. 2015. Hawkes processes in finance. *Market Microstructure and Liquidity* 1(1).

Borrajo, D.; Veloso, M.; and Shah, S. 2020. Simulating and classifying behavior in adversarial environments based on action-state traces: an application to money laundering. In *Proceedings of the 1st ACM International Conference on AI in Finance*.

Chang, X.; Liu, X.; Wen, J.; Li, S.; Fang, Y.; Song, L.; and Qi, Y. 2020. Continuous-Time Dynamic Graph Learning via Neural Interaction Processes. In *Proceedings of the 2020 ACM on Conference on Information and Knowledge Management*.

Cressie, N.; and Wikle, C. K. 2011. *Statistics for Spatio-Temporal Data*. ISBN 978-0-471-69274-4.

Daley, D.; and Vere-Jones, D. 2003. An introduction to the theory of point processes. Vol. I: Elementary theory and methods. 2nd ed Vol. 1. doi:10.1007/b97277.

Denton, E.; and Fergus, R. 2018. Stochastic Video Generation with a Learned Prior. volume 80 of *Proceedings of Machine Learning Research*, 1174–1183. Stockholmsmässan, Stockholm Sweden: PMLR.

Diggle, P. 2014. *Statistical Analysis of Spatial and Spatio-Temporal Point Patterns*. New York: Chapman and Hall/CRC. doi:doi.org/10.1201/b15326.

Du, N.; Dai, H.; Trivedi, R.; Upadhyay, U.; Gomez-Rodriguez, M.; and Song, L. 2016. Recurrent Marked Temporal Point Processes: Embedding Event History to Vector. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, 1555–1564. New York, NY, USA: Association for Computing Machinery. ISBN 9781450342322. doi: 10.1145/2939672.2939875.

Dumoulin, V.; and Visin, F. 2016. A guide to convolution arithmetic for deep learning. *arXiv e-prints* arXiv:1603.07285.

Farajtabar, M.; Du, N.; Gomez Rodriguez, M.; Valera, I.; Zha, H.; and Song, L. 2014. Shaping Social Activity by Incentivizing Users. In Ghahramani, Z.; Welling, M.; Cortes, C.; Lawrence, N.; and Weinberger, K. Q., eds., *Advances in Neural Information Processing Systems*, volume 27, 2474–2482. Curran Associates, Inc.

Farajtabar, M.; Wang, Y.; Gomez-Rodriguez, M.; Li, S.; Zha, H.; and Song, L. 2016a. COEVOLVE: A Joint Point Process Model for Information Diffusion and Network Co-evolution. In *Advances in Neural Information Processing Systems*, volume 29.

Farajtabar, M.; Ye, X.; Harati, S.; Song, L.; and Zha, H. 2016b. Multistage Campaigning in Social Networks. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, NIPS'16, 4725–4733. Red Hook, NY, USA: Curran Associates Inc. ISBN 9781510838819.

Franceschi, J.-Y.; Delasalles, E.; Chen, M.; Lamprier, S.; and Gallinari, P. 2020. Stochastic Latent Residual Video Prediction. *arXiv e-prints* arXiv:2002.09219.

Gerdes, G.; Greene, C.; Liu, X.; ; and Massaro, E. 2020. The 2019 Federal Reserve Payments Study. Technical report, Board of Governors of the Federal Reserve System.

González, J. A.; Rodríguez-Cortés, F. J.; Cronie, O.; and Mateu, J. 2016. Spatio-temporal point process statistics: A review. *Spatial Statistics* 18: 505 – 544. ISSN 2211-6753. doi:10.1016/j.spasta.2016.10.002.

Goodfellow, I.; Bengio, Y.; and Courville, A. 2016. *Deep Learning*. MIT Press. http://www.deeplearningbook.org.

Goyal, P.; Kamra, N.; He, X.; and Liu, Y. 2017. DynGEM: Deep embedding method for dynamic graphs. In *Proceedings of IJCAI 2017 Workshop on Representation Learning for Graphs*.

Goyal, P.; Kamra, N.; He, X.; and Liu, Y. 2020. dyngraph2vec: Capturing network dynamics using dynamic graph representation learning. *Knowledge-Based Systems* 187.

Gregor, K.; Papamakarios, G.; Besse, F.; Buesing, L.; and Weber, T. 2019. Temporal Difference Variational Auto-Encoder. In *International Conference on Learning Representations*.

Hawkes, A. G. 1971. Spectra of Some Self-Exciting and Mutually Exciting Point Processes. *Biometrika* 58(1): 83–90. ISSN 00063444.

He, J.; Lehrmann, A.; Marino, J.; Mori, G.; and Sigal, L. 2018. Probabilistic Video Generation Using Holistic Attribute Control. In Ferrari, V.; Hebert, M.; Sminchisescu, C.; and Weiss, Y., eds., *Computer Vision – ECCV 2018*, 466–483. Cham: Springer International Publishing. ISBN 978-3-030-01228-1.

Hochreiter, S.; and Schmidhuber, J. 1997. Long Short-Term Memory. *Neural Comput.* 9(8): 1735–1780. ISSN 0899-7667. doi:10.1162/neco.1997.9.8.1735.

Ioannidis, V. N.; Romero, D.; and Giannakis, G. B. 2018. Inference of Spatio-Temporal Functions Over Graphs via

Multikernel Kriged Kalman Filtering. *IEEE Transactions on Signal Processing* 66(12): 3228–3239. doi:10.1109/TSP.2018.2827328.

Isham, V.; and Westcott, M. 1979. A self-correcting point process. *Stochastic Processes and their Applications* 8(3): 335 – 347. ISSN 0304-4149. doi:https://doi.org/10.1016/0304-4149(79)90008-5.

Kingma, D. P.; and Welling, M. 2013. Auto-Encoding Variational Bayes. *arXiv e-prints* arXiv:1312.6114.

Kingma, D. P.; and Welling, M. 2019. An Introduction to Variational Autoencoders. *Foundations and Trends® in Machine Learning* 12(4): 307–392. ISSN 1935-8237. doi:10.1561/2200000056.

Kipf, T. N.; and Welling, M. 2016. Variational Graph Auto-Encoders. *NIPS Workshop on Bayesian Deep Learning* .

Li, J.; Dani, H.; Hu, X.; Tang, J.; Chang, Y.; and Liu, H. 2017. Attributed Network Embedding for Learning in a Dynamic Environment. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*.

Lian, W.; Henao, R.; Rao, V.; Lucas, J.; and Carin, L. 2015. A Multitask Point Process Predictive Model. volume 37 of *Proceedings of Machine Learning Research*, 2030–2038. Lille, France: PMLR.

Mehrasa, N.; Jyothi, A. A.; Durand, T.; He, J.; Sigal, L.; and Mori, G. 2019. A Variational Auto-Encoder Model for Stochastic Point Processes. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 3160–3169. doi:10.1109/CVPR.2019.00328.

Mei, H.; and Eisner, J. 2017. The Neural Hawkes Process: A Neurally Self-Modulating Multivariate Point Process. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, 6757–6767. Red Hook, NY, USA: Curran Associates Inc. ISBN 9781510860964.

Nguyen, G. H.; Lee, J. B.; Rossi, R. A.; Ahmed, N. K.; Koh, E.; and Kim, S. 2018. Continuous-time dynamic network embeddings. In *Proceedings of the The Web Conference 2018*.

Pareja, A.; Domeniconi, G.; Chen, J.; Ma, T.; Suzumura, T.; Kanezashi, H.; Kaler, T.; Schardl, T. B.; and Leiserson, C. E. 2020. EvolveGCN: Evolving Graph Convolutional Networks for Dynamic Graphs. In *Proceedings of the 34th AAAI Conference on Artificial Intelligence*.

Reinhart, A. 2018. A Review of Self-Exciting Spatio-Temporal Point Processes and Their Applications. *Statistical Science* 33(3): 299–318. ISSN 0883-4237. doi:10.1214/17-sts629. URL http://dx.doi.org/10.1214/17-STS629.

Rotatori, D. 2018. Real-time payments are changing the reality of payments. Technical report, Deloitte.

Salmon, D. 2019. Venmo Transaction Dataset. URL https://github.com/sa7mon/venmo-data.

Seo, Y.; Defferrard, M.; Vandergheynst, P.; and Bresson, X. 2018. Structured Sequence Modeling with Graph Convolutional Recurrent Networks. In *Proceedings of the 2018 International Conference on Neural Information Processing*.

Simonovsky, M.; and Komodakis, N. 2018. GraphVAE: Towards Generation of Small Graphs Using Variational Autoencoders. *CoRR* abs/1802.03480.

Trivedi, R.; Dai, H.; and Song, L. 2017. Know-evolve: deep temporal reasoning for dynamic knowledge graphs. In *Proceedings of the 34th International Conference on Machine Learning*.

Trivedi, R.; Farajtabar, M.; Biswal, P.; and Zha, H. 2019. DyRep: Learning Representations Over Dynamic Graphs. In *Proceedings of the 7th International Conference on Learning Representations*.

Venmo. 2020. Example Venmo Payment Feed. URL http://venmo.com.

Weiss, J. C.; and Page, D. 2013. Forest-Based Point Process for Event Prediction from Electronic Health Records. In Blockeel, H.; Kersting, K.; Nijssen, S.; and Železný, F., eds., *Machine Learning and Knowledge Discovery in Databases*, 547–562. Berlin, Heidelberg: Springer Berlin Heidelberg. ISBN 978-3-642-40994-3.

Xiao, S.; Farajtabar, M.; Ye, X.; Yan, J.; Song, L.; and Zha, H. 2018. Wasserstein Learning of Deep Generative Point Process Models. In *Advances in Neural Information Processing Systems*, volume 31.

Xiao, S.; Yan, J.; Yang, X.; Zha, H.; and Chu, S. M. 2017. Modeling the Intensity Function of Point Process via Recurrent Neural Networkss. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, AAAI'17, 1597–1603. AAAI Press.

Xu, H.; Chen, W.; Zhao, N.; Li, Z.; Bu, J.; Li, Z.; Liu, Y.; Zhao, Y.; Pei, D.; Feng, Y.; Chen, J.; Wang, Z.; and Qiao, H. 2018. Unsupervised Anomaly Detection via Variational Auto-Encoder for Seasonal KPIs in Web Applications. In *Proceedings of the 2018 World Wide Web Conference*, WWW '18, 187–196. Republic and Canton of Geneva, CHE: International World Wide Web Conferences Steering Committee. ISBN 9781450356398. doi:10.1145/3178876.3185996.

Yu, B.; Yin, H.; and Zhu, Z. 2018. Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence*.

Zipkin, J. R.; Schoenberg, F. P.; Coronges, K.; and Bertozzi, A. L. 2016. Point-process models of social network interactions: Parameter estimation and missing data recovery. *European Journal of Applied Mathematics* 27(3): 502–529. doi:10.1017/S0956792515000492.

Zuo, Y.; Liu, G.; Lin, H.; Guo, J.; Hu, X.; and Wu, J. 2018. Embedding Temporal Network via Neighborhood Formation. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*.