Churn Modeling with Probabilistic Meta Paths-based Representation Learning

Sandra Mitrović^{a,*}, Jochen De Weerd^a

^a*Department of Decision Sciences and Information Management, KU Leuven, Leuven, Belgium*

Abstract

Finding structural and efficient ways of leveraging available data is not an easy task, especially when dealing with network data, as is the case in telco churn prediction. Several previous works have made advancements in this direction both from the perspective of churn prediction, by proposing augmented call graph architectures, and from the perspective of graph featurization, by proposing different graph representation learning methods, frequently exploiting random walks. However, both graph augmentation as well as representation learning-based featurization face drawbacks. In this work, we first shift the focus from a homogeneous to a heterogeneous perspective, by defining different probabilistic meta paths on augmented call graphs. Secondly, we focus on solutions for the usually significant number of random walks that graph representation learning methods require. To this end, we propose a sampling method for random walks based on a combination of most suitable random walk generation strategies, which we determine with the help of corresponding Markov models. In our experimental evaluation, we demonstrate the benefits of probabilistic meta path-based walk generation in terms of predictive power. In addition, this paper provides promising insights regarding the interplay of the type of meta path and the predictive outcome, as well as the potential of sampling random walks based on the meta path structure in order to alleviate the computational requirements of representation learning by reducing typically sizable required data input.

Keywords: Representation Learning, Heterogeneous Networks, Meta Path-based

*Corresponding author

Email address: sandra.mitrovic@kuleuven.be (Sandra Mitrović)

1. Introduction

In churn prediction, just as in any other predictive analytics problem, there is a need to effectively leverage available data and provide more structured and efficient ways of handling them. This is especially important when dealing with complex data of large size, as is the case for churn prediction where Call Detailed Records (CDRs) are typically represented in the form of large call graphs. For structural and automated feature extraction, representation learning (Bengio et al., 2003) has been frequently used in recent times, primarily for text (Bengio et al., 2013; Mikolov et al., 2013a) but also for other data types. As such, several works have proposed different ways of extending existing text representation learning methods to graphs. A significant part of this stream of work is based on generating random walks and enforcing representations of nearby nodes to be close to each other (Perozzi et al., 2014; Grover & Leskovec, 2016), similarly to the (distributional) context sharing approach in the text domain. Along these lines, and to provide a more structural way of featurizing CDR graphs (both from perspective of avoiding feature hand-crafting and also better leveraging of available data), in Mitrović et al. (2017); Mitrović et al. (2019), we have proposed a novel method based on a combination of augmented CDR graphs and scalable node representation learning applied to these graphs. More precisely, the aforementioned augmented CDR graphs are constructed by deriving artificial nodes based on the RFM (**R**ecency, **F**requency, **M**onetary) model (Hughes, 1994) and connecting them to customers present in the original CDR graph, as appropriate. The augmented graphs are then used to create random walks based on which node representations are learned, as described above. The main drawback of the approach in Mitrović et al. (2017); Mitrović et al. (2019) is that random walks are generated considering the constructed augmented graphs as homogeneous, even though they contain different node types, which makes them inherently heterogeneous (Sun et al., 2011). Since different node types convey diverse information, previous studies have attempted to account for this heterogeneous information and have already demonstrated improvements in predictive accuracy for classification (and

clustering) problems (Sun et al., 2011; Dong et al., 2017; Fu et al., 2017). Hence, guiding random walks in order to generate more specifically tailored contexts to account for (and compare) these differences, seems worthwhile. Therefore, in this work, we exploit the same kind of networks as in Mitrović et al. (2017); Mitrović et al. (2019) but we adjust the representation learning part to accommodate for heterogeneous networks, using meta paths, similarly as has been done in Dong et al. (2017). However, unlike the meta paths defined in Dong et al. (2017), we allow for probabilistic and asymmetrical meta paths.

Another drawback of existing representation learning methods on graphs lies in the number of random walks needed to perform learning. First, the number of walks is typically a parameter which varies from one study to another and is either borrowed from previous works (Cao et al., 2016; Nguyen et al., 2018) or is arbitrarily set (Grover & Leskovec, 2016). Second, and more importantly, this number is usually very high (up to 80 walks per node in Perozzi et al. (2014)) which, especially in case of large graphs, requires substantial amount of computational time. Therefore, in this work, we additionally propose to take into account only a sample of random walks so as to reduce the amount of data necessary for representation learning. Furthermore, as we aim to sample the most informative random walks, we propose to use Markov models reflecting different meta path types. More specifically, given a large set of random walks, we aim to sample those walks which are most in line with one or more Markov models. In this work, the used Markov models are directly related to the types of meta paths. However, the method can be generalized to for instance expert-driven Markov model specification or alternative Markov model specifications. In our experimental evaluation, we assess the performance of such a Markov model-based sampling at node-level (thus sampling walks per node) in comparison to standard representation learning using different numbers of walks.

Our results show that representation learning on heterogeneous graphs using different meta paths can be beneficial for churn prediction as compared to representation learning on homogeneous graphs. Moreover, using different types of meta paths leads to interesting insights which provide motivation for further exploration of the interplay between different interactions, corresponding meta paths, and the predictive result. With

respect to sampling random walks, our results indicate that smart sampling of random walks is an interesting approach. Nevertheless, a more profound analysis and tuning of our proposed method are required so as to provide a fully robust alternative to the “use-as-much-data-as-possible” approach.

The rest of the paper is organized as follows: in Sections 2 and 3, we provide a short overview of related literature and preliminaries, respectively; in Section 4 we elaborate our methodology, while in Section 5 we explain our experimental evaluation; Section 6 provides the results of our experiments, Section 7 discusses them, and finally, Section 8 concludes our study with several directions for future research.

2. Related Work

In this section, a short overview of existing representation learning-based literature is provided, from two different perspectives: with respect to problem domain (churn prediction) and data type used (heterogeneous graphs).

Churn prediction in telco has so far been addressed in the literature from different perspectives, using different feature types and models. In this overview we limit ourselves mainly to representation learning-based methods. These methods have initially achieved great success for word embeddings in the natural language processing (NLP) domain (Bengio et al., 2003, 2013; Mikolov et al., 2013b), but were recently transferred to graphs primarily by Perozzi et al. (2014) but subsequently also by Tang et al. (2015b); Grover & Leskovec (2016); Cao et al. (2016); Dong et al. (2017). For the purpose of predicting churn in telco, explicit learning of customer representations was very rarely used. However, deep learning methods were used in several studies (Castanedo et al., 2014; Wangperawong et al., 2016; Umayaparvathi & Iyakutti, 2017) although not with too much success apart from the study by Castanedo et al. (2014), which however does not reveal all the details of the dataset used.

The idea of this work is to take advantage of the synergy of two existing works, i.e. an approach to apply explicit customer representation learning to churn prediction in Mitrović et al. (2017); Mitrović et al. (2019), and the work of Dong et al. (2017), which proposes to use meta paths for representation learning in heterogeneous graphs.

In Mitrović et al. (2017); Mitrović et al. (2019), an adaptation of node2vec (Grover & Leskovec, 2016) was developed to make it scalable on large graphs obtained from CDR data. Node2vec itself extends on DeepWalk (Perozzi et al., 2014) by defining more flexible random walks, which, however, come with an additional computational burden. Furthermore, in Mitrović et al. (2017); Mitrović et al. (2019) a solution is proposed for several problems known in the churn prediction domain, such as feature hand-crafting and under-exploiting of available data, as is the case in e.g. Chen et al. (2012); Kim et al. (2014). Given that augmented graph architectures proposed in Mitrović et al. (2017); Mitrović et al. (2019) enable integrating both information related to customer interaction as well as information conveyed by pure graph topology, we resort to using the same graph constructs. However, RFM-Augmented graphs in Mitrović et al. (2017); Mitrović et al. (2019) are considered homogeneous, while they are inherently heterogeneous in nature. Therefore, the option of using a meta path based representation learning method proposed by Dong et al. (2017) seems attractive. The latter method enables guiding random walks and is more appropriate for graphs containing different node types like collaboration graphs (with authors, papers and venue as nodes). Still, we slightly adapt it to our setting by defining a probabilistic version of random walks and not requiring walk symmetry.

Several other relevant works on heterogeneous networks have been proposed (Sun et al., 2011; Chang et al., 2015; Tang et al., 2015a; Chen & Sun, 2017; Fu et al., 2017; Jiang et al., 2017; Shi et al., 2018; Yang et al., 2018). The work by Sun et al. (2011) is also based on meta paths but it focuses on defining a similarity measure instead of learning representations. In Chang et al. (2015) learning representations is done by deep convolutional neural networks, while in Tang et al. (2015a) the focus is on text embeddings. Similar to our work, Chen & Sun (2017) and Fu et al. (2017) leverage particular meta paths in heterogeneous networks based on the predictive goal. However, the particularities of both methods differ from our work. Moreover, in Chen & Sun (2017) the goal is author identification while in Fu et al. (2017) learning is performed not only on the node, but also on the link level, which is not of interest for this work. Our work differs from one of Jiang et al. (2017) in that they first use all random walks to obtain scores which they then combine. Additionally, their focus is on the semi-

supervised setting. Shi et al. (2018) proposes learning different representations per node to address different aspects of nodes encapsulated within the heterogeneous network which is an interesting idea but is out of the scope of this paper. The idea of combining and reducing heterogeneous information is also presented in Yang et al. (2018) but they focus on meta graphs rather than meta paths and use deep denoising autoencoders for unsupervised dimensionality reduction.

3. Preliminaries

In this section, we will first provide several definitions which will be useful for the subsequent discussion of our methodology. Next, we provide a brief explanation of the construction of call networks from CDRs, as done in Mitrović et al. (2017); Mitrović et al. (2019).

3.1. Basic Definitions

Definition 3.1. (*RFM Model*) *The RFM Model as proposed by Hughes (1994) explains customer behavior in an observed time period with respect to a certain event based on the following three variables:*

- *Recency, which quantifies how recent the last event was of a customer during the observed time interval;*
- *Frequency, which quantifies how frequently a customer experienced a particular event during the observed time interval;*
- *Monetary, which quantifies the monetary amount a customer spent during the observed time interval in relation to this particular event.*

In our context, the particular event is a phone call but many other interpretations such as a purchase, a credit card transaction, etc. are possible, depending on the particular domain.

Definition 3.2. (*Heterogeneous network*) *A graph $G = (V, E)$ is called a heterogeneous graph if there exists a node type mapping function $\phi : V \rightarrow T_V$ and an edge type*

mapping function $\psi : E \rightarrow T_E$ such that either $|T_V| > 1$ or $|T_E| > 1$. We will use notation $G = (V, E, T_V, T_E, \phi, \psi)$ for such a graph.

Definition 3.3. (Probabilistic meta path) A path class $\mathcal{P}$ constructed over a heterogeneous graph $G = (V, E, T_V, T_E, \phi, \psi)$ which allows graph traversal through different node and edge types and can be written in the form $T_{v_1} \xrightarrow{T_{e_1}} \dots \xrightarrow{T_{e_{i-1}}} T_{v_i} \xrightarrow{T_{e_i}} T_{v_j} \xrightarrow{T_{e_j}} \dots \xrightarrow{T_{e_{k-1}}} T_{v_k}$ where $T_{v_i}, T_{v_j} \in T_V, T_{e_i}, T_{e_j} \in T_E$, with the transition probability at step i defined as:

$$\text{Prob}(T_{v_j} | T_{v_i}, \mathcal{P}) = \begin{cases} p_{\text{reg}}, & \text{if } T_{v_j} \neq T_{v_i} \\ p_{\text{min}}, & \text{otherwise} \end{cases}$$

where $p_{\text{min}} \ll p_{\text{reg}}$, is called a probabilistic meta path. In other words, we define the types of nodes which should be visited along the path with certain (high) probability p_{reg} , but with a very small probability p_{min} we allow diverging from the set path.

Definition 3.4. (Probabilistic meta path instance) A path $P \in \mathcal{P}$ of the form $v_1 \xrightarrow{e_1} \dots \xrightarrow{e_{i-1}} v_i \xrightarrow{e_i} v_j \xrightarrow{e_j} \dots \xrightarrow{e_{k-1}} v_k$ constructed over a heterogeneous graph $G = (V, E, T_V, T_E, \phi, \psi)$, where $v_i \in V, e_i \in E, \phi(v_i) = T_{v_i}, \phi(v_j) = T_{v_j}, \psi(e_i) = T_{e_i}, \psi(e_j) = T_{e_j}, T_{v_i} \neq T_{v_j}$ and $T_{e_i} \neq T_{e_j}$ is called an instance of probabilistic meta path $\mathcal{P}$ (a path following probabilistic meta path $\mathcal{P}$). In cases when the edge type is not crucial, a notation $P = (v_1, \dots, v_i, v_j, \dots, v_k)$ will be used.

Previous works (e.g. Dong et al. (2017)) define meta paths¹ to be symmetric ($T_{v_1} = T_{v_k}$) and without the possibility of taking alternate paths (hence $p_{\text{min}} = 0$). Moreover, to accommodate for particularities of the data and problem at hand, we will allow for probabilistic meta paths where T_{v_j} is not necessarily different from T_{v_i} (but the alternative option would still be possible only with a very small probability). The main motivation for introducing probabilistic meta paths is to ensure existence of meta path instances of sufficient length (as this might be a problem due to graph sparsity). In the

¹Previous works also use notion of relation instead of edge type which turns a meta path into a composite relation of different relation types (Sun & Han, 2013; Dong et al., 2017).

rest of the paper, we always refer to a probabilistic meta path even if we often use meta path for better readability.

3.2. Constructing RFM-Augmented Networks from Telco CDRs

In our earlier work Mitrović et al. (2017); Mitrović et al. (2019), a special type of call graph architectures, referred to as RFM-Augmented graphs have been constructed and their benefits for churn prediction in telco have been proven as compared to traditional RFM variables. We, therefore, opt for the same call graph architectures and provide a brief explanation of these.

The main idea of RFM-Augmented graphs is to augment the original call graph with artificial nodes which are generated based on RFM variables. Hence, it is important to define, first, how these RFM variables are calculated and second, how artificial nodes are generated based on those variables.

3.2.1. The Choice of RFM Variables

Despite the presence of a variety of RFM variables in the literature, to this end, only three different versions of RFM variables are considered: 1) Summary-RFM (denoted by RFM_s) whereby RFM variables are calculated considering all customer interactions; 2) Detailed-RFM (denoted by RFM_d), whereby customer interactions are divided according to call direction and destination into three subcategories: a) outgoing towards home network, b) outgoing towards other networks and c) incoming and each of R, F, M is calculated according to these; and 3) Churn-RFM (denoted by RFM_{ch}) whereby RFM variables are calculated only with respect to interactions with customers who are known to have churned. The latter is motivated by the fact that previous works on churn prediction in telco have recognized the importance of connections with already churned customers (Dasgupta et al., 2008; Modani et al., 2013; Zhang et al., 2012).

3.2.2. Generation of Artificial Nodes

Once the R/F/M variables are derived for each node, each of these R/F/M variables is partitioned in five groups (corresponding to very high, high, medium, low, very low) similar to procedures already used in the literature (Hughes, 1994; McCarty & Hastak, 2007; Cheng & Chen, 2009). Each of these groups is then assigned one artificial

node. In case of Summary-RFM (RFM_s), given that each node is assigned one variable per R, F, M, described partitioning results in 15 artificial nodes R_i, F_i, M_i , where $i \in \{1, 2, \dots, 5\}$. Next, nodes from the original call graph are connected to artificial nodes according to their corresponding R/F/M partitions. Such an architecture is denoted by AG_s . A similar set of steps is used to construct the second type of architecture, except that the procedure does not start with Summary-RFM variables, i.e. one variable per R, F, M (per node), but instead with Detailed-RFM (RFM_d) hence, three different variables per each of R, F, M (per node) which are further on partitioned (each of them separately) into five groups. This graph construction is denoted by AG_d . An additional augmentation of both of the previous two graph types is obtained by adding an artificial churn node to which all identified churners are then connected. The obtained RFM-Augmented graphs are denoted by AG_{s+ch} and AG_{d+ch} for AG_s and AG_d graph, respectively.

An example of a RFM-Augmented AG_{s+ch} network can be seen in Figure 1.

Figure 1: An illustration of the AG_{s+ch} RFM-Augmented network. The central part denoted by gray circles represents the original topology while surrounding nodes denoted by squares, diamonds and octagons represent artificial recency, frequency and monetary nodes, respectively. Finally, the node denoted CH inside a circle represents the artificial churn node to which all previously identified churners are connected.

4. Methodology

In this section, we first explain the construction of different types of guided random walks, through the concept of meta path (types). Next, we proceed with describing the derivation of Markov models and mixture Markov model-based random walk sampling. Finally, we elaborate on the representation learning procedure which is applied on sampled random walks.

4.1. Defining Meta Paths in RFM-Augmented Networks

In our previous work (Mitrović et al., 2017; Mitrović et al., 2019), we emphasized the presence of different types of nodes and links in RFM-Augmented graphs and use that as motivation for considering these graphs as unweighted. However, in the ensuing step, the representation learning method based on random walks does not make a distinction between different node types. Hence, essentially our previous method treats RFM-Augmented graphs as homogeneous. Nevertheless, by looking into Definition 3.2 we can see that these graphs are inherently heterogeneous, with $|T_V| = |T_E| = 2$ for AG_s and AG_d and $|T_V| = |T_E| = 3$ for AG_{s+ch} and AG_{d+ch} (for AG_s and AG_d , we can perceive that the nodes belonging to the original topology are connected by edges of one type, while the same nodes are connected to the artificial ones with another type of edges).

We will use the common notation $AG_* = (V, E, T_V, T_E, \phi, \psi)$ for all RFM-Augmented graphs, where $* \in \{s, s+ch, d, d+ch\}$ and $T_V = \{CN \cup AN\}$ with CN denoting the set of customer nodes and AN denoting the set of all artificial nodes. Additionally, in graphs AG_{s+ch} and AG_{d+ch} , we will make a distinction between a set of artificial RFM nodes, denoted as AN_{rfm} , and a set containing the single artificial churn node, denoted as AN_{ch} . Hence, $AN = AN_{rfm} \cup AN_{ch}$. As paying special attention to edge types (T_E) will not be necessary, we will further shorten the notation to $AG_* = (V, E, \{CN \cup AN\}, \phi)$ where $* \in \{s, d\}$ and $AG_* = (V, E, \{CN \cup AN_{rfm} \cup AN_{ch}\}, \phi)$, for $* \in \{s+ch, d+ch\}$.

In heterogeneous networks, on the contrary to homogeneous ones, nodes can be connected via different types of paths, known as meta paths defined in Definition 3.3.

Moreover, different node and edge types convey diverse information and taking that explicitly into account can enhance predictive performance, as shown in previous works, e.g. by Sun et al. (2011). Therefore, in order to leverage RFM-Augmented graphs, it is reasonable to define different types of probabilistic meta paths, as follows:

- A meta path $\mathcal{P}_C$ constructed over an RFM-Augmented graph $AG_* = (V, E, \{CN \cup AN\}, \phi)$ of the form $CN \rightarrow CN \rightarrow CN \rightarrow \dots$. Hence, this meta path discriminates in favor of the CN node type at each step (to be referred to as *local* meta path). In other words, we consider meta path instances $P_C \in \mathcal{P}_C$ of the form $P_C = (v_1, \dots, v_i, v_{i+1}, \dots, v_k)$, $(v_i, v_{i+1}) = e_i \in E$ where transition probability at step i is defined as:

$$Prob(v_{i+1}|v_i, P_C) = \begin{cases} 0.99, & \text{if } \phi(v_{i+1}) = CN \\ 0.01, & \text{if } \phi(v_{i+1}) = AN \end{cases}$$

- A meta path $\mathcal{P}_A$ constructed over an RFM-Augmented graph $AG_* = (V, E, \{CN \cup AN\}, \phi)$ of any of the two forms: $CN \rightarrow AN \rightarrow CN \rightarrow \dots$ or $AN \rightarrow CN \rightarrow AN \rightarrow \dots$. Hence, this meta path favors alternation between CN and AN node types at each step (to be referred to as *alternating* meta path). In other words, we consider meta path instances $P_A \in \mathcal{P}_A$ of the form $P_A = (v_1, \dots, v_i, v_{i+1}, \dots, v_k)$, $(v_i, v_{i+1}) = e_i \in E$ where transition probability at step i is defined as:

$$Prob(v_{i+1}|v_i, P_A) = \begin{cases} 0.99, & \text{if } \phi(v_i) \neq \phi(v_{i+1}) \\ 0.01, & \text{otherwise} \end{cases}$$

- A meta path $\mathcal{P}_{\bar{C}}$ constructed over an RFM-Augmented graph $AG_* = (V, E, \{CN \cup AN_{rfm} \cup AN_{ch}\}, \phi)$ of the form $CN \rightarrow AN_{ch} \rightarrow CN \rightarrow \dots$. Hence, this meta path discriminates in favor of the artificial churn node AN_{ch} after each CN node type, whenever possible, that is, if such link exists. This meta path will be referred to as *local-churn* meta path. In other words, we consider meta path instances $P_{\bar{C}} \in \mathcal{P}_{\bar{C}}$ of the form $P_{\bar{C}} = (v_1, \dots, v_i, v_{i+1}, \dots, v_k)$, $(v_i, v_{i+1}) = e_i \in E$ where transition prob-

ability at step i is defined as:

$$Prob(v_{i+1}|v_i, \mathcal{P}_{\bar{C}}) = \begin{cases} 0.90, & \text{if } \phi(v_{i+1}) = AN_{ch} \\ 0.09, & \text{if } \phi(v_{i+1}) = CN \\ 0.01, & \text{if } \phi(v_{i+1}) = AN_{rfm} \end{cases}$$

- A meta path $\mathcal{P}_{\bar{A}}$ constructed over an RFM-Augmented graph $AG_* = (V, E, \{CN \cup AN_{rfm} \cup AN_{ch}\}, \phi)$ of any of the two forms: $CN \rightarrow AN_{ch} \rightarrow CN \rightarrow AN_{rfm} \rightarrow CN \rightarrow \dots$ or $AN_{rfm} \rightarrow CN \rightarrow AN_{ch} \rightarrow CN \rightarrow AN_{rfm} \rightarrow CN \rightarrow \dots$. Hence, this meta path favors visiting churn node type AN_{ch} while alternating between CN and AN node types, whenever possible, that is, if such link exists. This meta path will be referred to as *alternating-churn* meta path. In other words, we consider meta path instances $P_{\bar{A}} \in \mathcal{P}_{\bar{A}}$ of the form $P_{\bar{A}} = (v_1, \dots, v_i, v_{i+1}, \dots, v_k)$, $(v_i, v_{i+1}) = e_i \in E$ where transition probability at step i is defined as:

$$Prob(v_{i+1}|v_i, \mathcal{P}_{\bar{A}}) = \begin{cases} 0.90, & \text{if } \phi(v_{i+1}) = AN_{ch} \\ 0.09, & \text{if } \phi(v_{i+1}) \neq AN_{ch} \\ & \text{and } \phi(v_i) \neq \phi(v_{i+1}) \\ 0.01, & \text{if } \phi(v_{i+1}) \neq AN_{ch} \\ & \text{and } \phi(v_i) = \phi(v_{i+1}) \end{cases}$$

Obviously, *-churn meta paths are not to be considered on AG_s and AG_d RFM-Augmented graphs.

Figure 2 depicts one fragment of an RFM-Augmented AG_{s+ch} network (in a)) along with path instances for local meta path (in b)), alternating meta path (in c)), local-churn meta path (in d)) and alternating-churn meta path (in e)).

4.2. Random Walks Sampling based on a Mixture of Two Markov Models

Existing representation learning random walk-based methods require the generation of high amounts of walks of a certain length. Performance tends to improve with an increasing number of walks (although the performance increase becomes less significant

Figure 2: Graphical illustration of defined meta paths on the AG_{s+ch} RFM-Augmented network. In a) a fragment of the network depicting the neighborhood of node v . Nodes of type CN are represented by solid line-edged circles. Nodes of type AN_{rfm} are denoted by circles with dotted edges. The artificial churn node (of type AN_{ch}) is a diamond with dotted edges. Connections with artificial nodes (AN) are dotted while edges interconnecting nodes of type CN are solid lines. In b)-e) instances of defined meta paths are represented (all starting from node v). Particularly: in b) an example of a local meta path; in c) an instance of an alternating meta path; in d) instance of a local-churn meta path and finally, in e) an instance of an alternating-churn meta path.

after a certain number of walks). Ideally, one would like to be able to discern interesting walks which contain enough variety to learn good representations even with reduced number of random walks. We approach this problem by sampling random walks using a specific Markov model-based procedure as explained below.

Lets us denote with X the random variable denoting the choice of a node v in random walk r constructed according to one of the meta path types defined in Section 4.1. Obviously, X is observable and takes discrete values from the set $\{CN \cup AN_{rfm} \cup AN_{ch}\}$. Additionally, note that each of devised meta path types defines the transition probability to the next node based exclusively on the current node without any further control imposed. Given all these, the stochastic process generating random variable X is a discrete, first order (observable) Markov chain (Drake, 1967), with three different states. Namely, we can assume that all observations where X takes values of one of the customer nodes originate from a state CN , all observations where X takes values of one of

the artificial *RFM* nodes originate from a state AN_{rfm} and finally observations where X is just the artificial churn node originate from a state denoted as AN_{ch} ².

Next, we can simplify generated random walks by performing observation coalescing based on node type, as described in Algorithm 1. Coalescing for customer nodes (line 5) and RFM artificial nodes (line 6) is straightforward as each of them has only to be substituted with a corresponding unified observation for its node type. Doing the same for artificial churn node does not make sense as we already have a unique artificial churn node. However, since the artificial churn node serves only to denote churning customers, for coalescing churners, we have to take into account surrounding customer nodes. More precisely, a customer node whose immediate neighbor in random walk is artificial churn node, should not be treated as standard customer node but instead as a churner (node). We, therefore, look into all triplets and tuples of observations in the random walk and perform coalescing if applicable (lines 9-18).

Algorithm 1 Coalescing of a Random Walk

Input: A random walk r , where $r = (... , X_{i-1}, X_i, X_{i+1}, ...)$, and X_i is an observation at time t

- 1: Let a, b, c be the new unifying observations for $X \in \{CN\}, \{AN_{rfm}\}, \{AN_{ch}\}$ respectively;
- 2: $\hat{r} := r$;
- 3: **for each** observation $X \in \hat{r}$ **do**:
- 4: **switch** X **do**
- 5: **case** $X \in CN$: Replace X with a in $\hat{r}$;
- 6: **case** $X \in AN_{rfm}$: Replace X with b in $\hat{r}$;
- 7: **end switch**
- 8: **end for**;
- 9: **for each** triplet of observations $(X_{i-1}, X_i, X_{i+1}) \in \hat{r}$ **do**:
- 10: **if** $(X_{i-1}, X_i, X_{i+1}) = (a, c, a)$ **then**:
- 11: Replace (X_{i-1}, X_i, X_{i+1}) with cc in $\hat{r}$;
- 12: **end if**
- 13: **end for**;
- 14: **for each** tuple of observations $(X_i, X_{i+1}) \in \hat{r}$ **do**:
- 15: **if** $(X_i, X_{i+1}) = (a, c)$ **then**:
- 16: Replace (X_i, X_{i+1}) with c in $\hat{r}$;
- 17: **end if**
- 18: **end for**;

Output: Coalesced random walk $\hat{r}$

²To ease readability, we use the same notation for Markov model states and node types. We believe that this does not hinder the clarity of the paper.

Next, the resulting coalesced random walks are used to learn the model, i.e. the transition probabilities (which might substantially differ from the starting transition probabilities). This kind of procedure can be performed to create a Markov model for all different meta path types.

The key idea of our approach resides in using only a small fraction of random walks, as compared to other random walk-based representation learning approaches which are generating a huge number of random walks (restarting from the same node several times (Perozzi et al., 2014; Grover & Leskovec, 2016; Dong et al., 2017)). In order to be able to learn useful representations from a small number of walks, these walks need to be informative. Therefore, we propose to consider a combination (sort of a mixture) of Markov models, based on different meta path types, instead of using one particular meta path type. More precisely, we impose using only one random walk per node per selected meta path type. The choice of the different meta path types to use can be done by scoring sequences generated by one model with another (for all available Markov models) and finally selecting top X generating the least likely sequences. Alternatively, in the case of a predictive problem, this can be done by simply choosing the two top performing ones (evaluated on a toy example). The walks themselves can be generated using the newly learned transition probabilities or domain expert defined ones.

4.3. Meta path based Representation Learning on RFM-Augmented Networks

As already explained in Section 2, our approach is similar to the metapath2vec approach proposed by Dong et al. (2017), which in turn is based on the SkipGram model first proposed in the NLP domain and the ideas of its adaptation to graph settings by Perozzi et al. (2014) and Grover & Leskovec (2016).

SkipGram itself is a likelihood optimization problem of finding a low (d -) dimensional representation f for each word v of vocabulary V such that the probability of predicting its nearby words (within its corresponding context C_v) is maximized. That is, the model attempts to find $f, f : V \rightarrow R^d, d \ll |V|$ such that

$$\max \sum_{v \in V} \sum_{w \in C_v} \log Pr(w|f(v)).$$

The background hypothesis of SkipGram states that words are more similar the more

they appear in similar contexts (known as a distributional hypothesis). Given that similar hypothesis exists in the graph setting in the form of homophily, the SkipGram idea can be transferred to graphs, by redefining the concept of word context into a concept of node neighborhood. Node neighborhood is usually defined by means of truncated random walks starting from the node itself. Moreover, in heterogeneous networks, we can use the concept of meta paths to define different types of context. Then the above objective becomes:

$$\max_f \sum_{v \in V} \sum_{w_P \in C_v^P} \log Pr(w_P | f(v)),$$

where C_v^P is a context of a node v defined by a meta path P . The conditional probability $Pr(w | f(v))$ usually defined by a soft-max function becomes $Pr(w_P | f(v)) = \frac{\exp(f(w_P) \cdot f(v))}{\sum_{u \in V} \exp(f(u) \cdot f(v))}$. Given the complexity of the denominator and the goal (which is essentially to find useful representations and not to find the exact optimum of the function), negative sampling is used to approximate the objective, thus leading to a new objective being

$$\max_f \sum_{w_P \in C_v^P} \log \sigma(-f(w_P) \cdot f(v)) + \sum_{n_P \notin C_v^P} \log \sigma(-f(n_P) \cdot f(v)),$$

where σ is the sigmoid function.

Node representations obtained by the previously explained procedure are used as input features for an l_2 -regularized logistic regression classifier aiming to predict churn.

5. Experimental Evaluation

5.1. Data and Tools

For our experimental evaluation, we use two datasets (one prepaid, one postpaid). Both datasets span across four months, have the same structure and represent the restricted version of CDR data, containing only the information about: caller, callee³, date/time and call duration. On the one hand, this limits the diversity of available information, but on the other hand, ensures the possibility for reproducing our method on other datasets as any telco operator would have the information used here at their disposal, in contrast to some other churn prediction approaches which tend to avoid

³Both caller and callee numbers are encrypted to respect customer privacy.

disclosure of the complete information used, like Castanedo et al. (2014), or opt for using not so easily available data e.g. customer complaints in Huang et al. (2015). The characteristics of monthly graphs constructed from these datasets are provided in Table 1. Our code is implemented in Python.

Table 1: Characteristics of the graphs obtained from the datasets. Notation: $|V|$ - number of nodes, $|E|$ - number of edges, $avg(deg)$ - average degree, $avg(cl.coeff.)$ - average clustering coefficient, $|CC|$ - number of connected components, $|max_cq|$ - size of maximal clique, dac - degree assortativity coefficient.

Dataset	$ V $	$ E $	$avg(deg)$	$avg(cl.coeff.)$	$ CC $	$ max_cq $	dac
Prepaid	4303541	5936423	2.75886	0.05749	138509	7	-0.00110
Postpaid	4799149	9246134	3.85324	0.06939	27392	7	-0.02290

5.2. Experimental Setup

For the first part of our analysis where we evaluate different meta path type-based representation learning approaches, we use two methods as baselines. One is based on logistic regression using traditional RFM variables calculated on customer level as features. The other is similar to our method in that it is the representation learning approach applied on our RFM-Augmented networks (AG_s , AG_d , AG_{s+ch} , AG_{d+ch}), except that instead of biasing random walks by different meta path types, a genuinely random walk, considering RFM-Augmented networks as homogeneous, is performed. We compare these with four different scenarios, based on four differently defined meta paths.

In order to apply our representation learning method, we opt for the following parameter settings: for random walks, the walk length is set to 30 and the number of walks to 10 and for the SkipGram model, the number of dimensions is set to 16, window length to 10 and the number of negative samples to 5.

In the absence of churn labels in original datasets (we have only the information of small number of ported out customers), labeling of churners is done based on the customer inactivity. More precisely, given the four months $M-1$, M , $M+1$, $M+2$, for customers who have called during month M , we aim at predicting their probability of churning during the first three weeks of month $M+2$. This leads to constructing the network for month M in which, following the same definition, we label as churners customers who had calls during month $M-1$ (and month M) but who neither had

calls during the first three weeks of month $M+1$ nor have been ported out during the same period. According to this churn definition, prepaid and postpaid datasets contain approximately 12.2% and 7.8% churners, respectively.

Defining churners based on inactivity is the most prevailing approach in the current literature. However, unlike most works which consider one month inactivity periods, we only use the first three weeks so as to make this setting meaningful for real-life application scenarios by allowing for the one remaining week to be used for actionable purposes such as retention campaigns.

For the second part, in which we evaluate different samples of random walks, we fix walk length to 30 and window length to 10, but experiment both with 16 and 32 dimensions.

6. Experimental Results

In this section we present the results of two analyses. The first one is related to the comparison of different meta path-based approaches with their baselines: classical RFM features and the homogeneous approach, i.e., the one which uniformly samples nodes (regardless their node type). The second analysis is related to benchmarking representation learning methods based on different number of random walks and meta path types in which we investigate how well our mixed Markov model-based sampling performs.

Results for the first analysis, obtained using different meta paths on both datasets and different kinds of RFM-Augmented networks can be seen in Table 2. It is obvious that an alternating meta path-based approach always outperforms a local-based approach, both in terms of AUC and lift (at 0.5%). Likewise, both in terms of AUC and lift, the alternating-churn meta path-based approach performs better than the local-churn one. Our local meta path definition reflects homophily⁴ based on customer interactions. Given that an alternating meta path reflects the factual similarity of customers measured by R, F, M variables, we can infer that homophily indeed plays an important role for predicting churn, which confirms findings of previous studies (Verbeke et al., 2014).

⁴Homophily is a tendency to link to those who are perceived as similar.

Table 2: Comparison in terms of AUC and lift (at 0.5%, between parenthesis) among different methods for prepaid (upper) and postpaid (lower) datasets. The results are averaged across 10 folds (different from the folds used for hyperparameter tuning). Baselines are marked with the star symbol. The best AUC score per RFM-Augmented graph type per dataset is marked in bold. The best overall AUC score per dataset is marked in bold and underlined.

Dataset	Graph Type	RFM Variables*	Homogeneous*	Heterogeneous: meta path-based			
				Local	Alternating	Local-churn	Alternating-churn
Prepaid	AG_s	0.671 (1.788)	0.669 (1.850)	0.571 (1.596)	0.680 (1.864)	n/a	n/a
	AG_d	0.683 (1.857)	0.665 (1.834)	0.554 (1.409)	0.673 (1.860)	n/a	n/a
	AG_{s+ch}	0.671 (1.789)	0.671 (1.875)	0.569 (1.397)	0.684 (1.889)	0.578 (1.684)	0.683 (1.913)
	AG_{d+ch}	0.682 (1.856)	0.666 (1.859)	0.554 (1.318)	0.676 (1.911)	0.570 (1.581)	0.678 (1.942)
Postpaid	AG_s	0.741 (3.367)	0.746 (3.409)	0.623 (2.587)	0.752 (3.656)	n/a	n/a
	AG_d	0.750 (3.750)	0.739 (3.360)	0.617 (2.379)	0.741 (3.769)	n/a	n/a
	AG_{s+ch}	0.741 (3.369)	0.746 (3.401)	0.622 (2.543)	0.752 (3.827)	0.613 (2.476)	0.751 (3.802)
	AG_{d+ch}	0.750 (3.751)	0.738 (3.331)	0.615 (2.456)	0.741 (3.768)	0.607 (2.401)	0.740 (3.856)

Additionally, we can notice that for AG_s and AG_{s+ch} , both in terms of AUC and lift, the learning approach based on alternating meta paths outperforms all the other methods for both datasets. Furthermore, alternating meta path combined with AG_{s+ch} is the overall winner across all network types and methods, for both datasets. Moreover, results of the non-parametric pairwise DeLong, DeLong, Clarke-Pearson statistical test (DeLong et al., 1988) showcase significant differences in AUC scores, at the 95% confidence level. More specifically, for prepaid, the highest AUC score is statistically significantly better than all other AUC scores except the one obtained for AG_s and alternate meta path, while for postpaid, the highest AUC score statistically significantly outperforms all the rest (see Table A.1 and Table A.2 in Appendix A for prepaid and postpaid, respectively).

Results for our analysis focused on benchmarking representation learning methods in terms of the number of random walks and meta path types can be seen in Table 3. We compare methods with random sampling of one, two and ten walks per two different random walk generation strategies (homogeneous and alternating meta path) with our approach of mixed sampling of one homogeneous plus one alternating meta path-based walk. As expected, using more walks increases performance although with adding more walks this increase reduces remarkably (evidenced from our own experimentations). However, the predictive performance benefit of using ten walks has a major drawback in the computational time needed for generating both walks and representations. Therefore, we rather opt for comparison with the method based on the same number of walks. As can be observed, our strategy of using two mixed walks performs similarly to the

one using two homogeneous walks on both datasets: on prepaid dataset it performs slightly sub-par to the alternate two walks approach, while for postpaid slightly outperforms the same method. It is worth noting that, for the mixed approach, more weight is imposed on alternating walks (0.8), than on homogeneous walks (0.2), given the gap in their performance. Despite this tuning it seems that this discrepancy in the individual AUC scores matters substantially. Ideally, we would be able to choose from different types of Markov models which have approximately the same performance since we attempt to apply a similar reasoning as with ensemble methods whereby combining several “weak” performers leads to better performance. Nevertheless, non-parametric pairwise DeLong, DeLong, Clarke-Pearson statistical test reported statistically significant differences when comparing the mixed approach both with homogeneous and alternate meta paths with two walks for both datasets (p-value=0.0<0.05 and p-value=1.3e-25<0.05, respectively for prepaid and p-value=0.0<0.05 and p-value=3.3e-33<0.05, respectively for postpaid), at the 95% confidence level (for 16 dimensions).

7. Discussion

In this section we would like to briefly discuss some aspects related to the two parts of our analysis.

First, regarding benchmarking different meta path-based approaches, we would like to discuss the effect of adding churn information on the obtained AUC scores. Our previous study indicates that switching from AG_s to AG_{s+ch} for random walk generation yields better AUC scores (see homogeneous baseline figures). On the one hand, our results based on both local and alternating heterogeneous meta paths, confirm this finding (compare results for AG_s vs. AG_{s+ch} and AG_d vs. AG_{d+ch}). On the other hand, if churn information indeed increases performance, we would expect that local-churn and alternating-churn meta path-based AUC scores outperform local and alternating ones, respectively. As this, however, does not seem to be the case, we performed a short comparison of generated local and local-churn meta path-based walks for prepaid AG_{s+ch} . The total number of generated walks in both cases is the same (43,035,570), however, the presence of AN_{ch} and AN_{rfm} nodes is remarkably different: with respect to total number of walks, AN_{rfm} nodes occur in 25.16% of walks in the case of local,

Table 3: Comparison in terms of AUC and lift (at 0.5%, between parenthesis) among number of walks and meta path types for prepaid (upper) and postpaid (lower) datasets on the example network of AG_{s+ch} . The results are averaged across 10 folds (different from folds used for hyperparameter tuning). The best AUC score per dataset and number of dimensions considering at most two walks is marked in bold. The best overall AUC score per dataset and number of dimensions is marked in bold and underlined.

Dataset	Dim.	10 walks		2 walks		1 walk		Mixed - 2 walks	
		Homog.	Alter.	Homog.	Alter.	Homog.	Alter.	Homog. + Alter.	
Prepaid	16	0.671 (1.875)	0.684 (1.889)	0.6457 (1.8216)	0.6748 (1.8742)	0.6428 (1.7003)	0.6716 (1.8521)	0.6737 (1.8403)	
	32	0.677 (1.922)	0.686 (2.032)	0.6508 (1.8489)	0.6747 (1.8522)	0.6548 (1.8423)	0.6749 (1.8919)	0.6739 (1.861)	
Postpaid	16	0.746 (3.401)	0.752 (3.827)	0.7076 (3.0854)	0.7472 (3.6096)	0.7002 (2.6355)	0.7428 (3.5834)	0.7483 (3.6335)	
	32	0.755 (3.486)	0.753 (3.779)	0.7148 (3.1338)	0.7472 (3.7532)	0.7189 (3.1075)	0.7434 (3.6489)	0.7483 (3.6111)	

and 24.20% in case of local-churn meta path-based walks, while AN_{ch} nodes occur in only 0.15% of walks in case of local and 19.99% of walks in case of local-churn meta path-based walks. Even though it might look counter-intuitive, it seems that the presence of AN_{rfm} contributes more to an improvement in AUC than AN_{ch} . A potential explanation could be that, given the small number of churners (only 10.92% out of observed customer base), forcing walks to visit AN_{ch} results in less coverage of the network which impedes the learning process, as opposed to visiting AN_{rfm} nodes which allow for reaching nodes with different characteristics. To further test this hypothesis, we experimented with the more strict version of the local meta path-approach to completely forbid visiting AN_{rfm} nodes (hence leaving 0.01 probability for AN_{ch} but setting probability for AN_{rfm} to 0). This resulted in further deterioration of the AUC score to 0.6044 (and lift to 1.5742).

Second, from Table 3 we can also notice that the performance depends not only on number of walks and meta path type, but the number of dimensions as well. It is obvious that with a decreasing number of dimensions, homogeneous meta paths are losing its predictive power at a higher rate than alternate meta path ones. Additionally, it is worth noting that the representation learning approach can always score better than traditional RFM variables, simply by increasing the number of dimensions. For example, even though from Table 2 might currently seem that RFM variables dominate for AG_d and AG_{d+ch} , increasing the number of dimensions to 128 yields AUC score of 0.754 for AG_d and 0.755 for AG_{d+ch} (postpaid dataset), thus outperforming traditional RFM variables. Additionally, it could be observed that a more diverse graph might benefit from a higher number of dimensions (and different number of dimensions might be optimal for different graph types).

Third, regarding the random walk sampling part and the combination method proposed, an obvious question arising is how to select the two targeted meta path types and how to generate Markov models based on these. With respect to the former, several different possibilities exist. For example, one could evaluate the dissimilarity of generated random walks based on a certain metric and select the Markov models generating the most dissimilar ones or make a predictive performance-driven decision. For this work, we opted for the latter, using the experimental results from the first part of our analy-

sis and generated (observed) random walks to derive actual transitional probabilities of generated models per each meta path type. This is one of the typical tasks related to Markov models. However, the choice of transition probabilities can be a domain-expert decision. In order to test the robustness of selected probabilities (that is, whether the deviation of transition probabilities remarkably impacts predictive results), we performed a test where we defined transition probabilities as follows:

For the homogeneous scenario:

$$P(CN|CN) = 0.5; P(AN_{rfm}|CN) = 0.499; P(AN_{dh}|CN) = 0.001$$

For the alternating meta path type:

$$P(CN|CN) = 0.1; P(AN_{rfm}|CN) = 0.899; P(AN_{dh}|CN) = 0.001$$

The obtained results are:

- For prepaid: AUC: 0.6728; lift: 1.8576
- For postpaid: AUC 0.7431; lift: 3.4688

Hence, we can claim that our method is robust to the choice of transition probabilities.

Besides the predictive performance, it is important to mention computational time savings when reducing the number of walks. More specifically, by taking into account only two walks instead of ten, an average speed up of almost 80% is observed on the prepaid dataset for the SkipGram part (walk generation could be assumed to take an equal amount of time as it is parallelizable).

We would like to mention that we have also experimented with random walks which instead of starting from a selected node, contain the selected node approximately in the middle (approximately as the current walk length is even). However, the results obtained were similar (if not worse) to the ones obtained when starting the node from each node, hence we opted for omitting them.

8. Conclusion

In this paper we make a synergy of existing advancements realized in the direction of leveraging networked data for telco churn prediction and probabilistic meta paths-based representation learning. The contributions of this work are three-fold.

First, we draw attention to the heterogeneous nature of previously proposed call graph extensions used for modeling churn in telco and devise probabilistic meta paths adapted for the data and problem at hand. As such, this is to the best of our knowledge, the first study to consider meta path based representation learning for churn prediction. Second, we perform an experimental analysis and show the benefits of applying representation learning on graphs using probabilistic meta paths, both in terms of the improvement in the predictive results as well as in promising insights regarding the interplay of meta path type and predictive outcome. Third, we propose a method for sampling random walks based on a mixture of Markov models in order to facilitate the tedious random walk-based representation learning approaches which require a large number of random walks. We demonstrate that this approach outperforms competing methods on at least one dataset. Additionally, this approach can be applied to any setting/problem where heterogeneous graphs are present and not only to churn prediction in Telco.

Apart from the sampling-related possibilities (such as weighting), a potential extension of this work can be the further refinement of node types in the obtained graphs that would consequently expand the space of potential meta paths. A more detailed study might be necessary to correlate different types of meta paths with social phenomena such as homophily and social influence as we see a lot of potential in this direction.

References

- Bengio, Y., Courville, A., & Vincent, P. (2013). Representation learning: A review and new perspectives. *IEEE Trans. Pattern Anal. Mach. Intell.*, 35, 1798–1828.
- Bengio, Y., Ducharme, R., Vincent, P., & Jauvin, C. (2003). A neural probabilistic language model. *Journal of machine learning research*, 3, 1137–1155.
- Cao, S., Lu, W., & Xu, Q. (2016). Deep neural networks for learning graph representations. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence AAAI’16* (pp. 1145–1152). AAAI Press.
- Castanedo, F., Valverde, G., Zaratiegui, J., & Vazquez, A. (2014). Using deep learning

- to predict customer churn in a mobile telecommunication network. URL: http://www.wiseathena.com/pdf/wa_dl.pdf.
- Chang, S., Han, W., Tang, J., Qi, G.-J., Aggarwal, C. C., & Huang, T. S. (2015). Heterogeneous network embedding via deep architectures. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining KDD '15* (pp. 119–128). New York, NY, USA: ACM.
- Chen, T., & Sun, Y. (2017). Task-guided and path-augmented heterogeneous network embedding for author identification. In *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining WSDM '17* (pp. 295–304). New York, NY, USA: ACM.
- Chen, Z.-Y., Fan, Z.-P., & Sun, M. (2012). A hierarchical multiple kernel support vector machine for customer churn prediction using longitudinal behavioral data. *European Journal of operational research*, 223, 461–472.
- Cheng, C.-H., & Chen, Y.-S. (2009). Classifying the segmentation of customer value via rfm model and rs theory. *Expert systems with applications*, 36, 4176–4184.
- Dasgupta, K., Singh, R., Viswanathan, B., Chakraborty, D., Mukherjea, S., Nanavati, A. A., & Joshi, A. (2008). Social ties and their relevance to churn in mobile telecom networks. In *Proceedings of the 11th conference on Extending database technology: Advances in database technology* (pp. 668–677). ACM.
- DeLong, E. R., DeLong, D. M., & Clarke-Pearson, D. L. (1988). Comparing the areas under two or more correlated receiver operating characteristic curves: a nonparametric approach. *Biometrics*, (pp. 837–845).
- Dong, Y., Chawla, N. V., & Swami, A. (2017). Metapath2vec: Scalable representation learning for heterogeneous networks. In *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 135–144). ACM.

- Drake, A. W. (1967). Discrete-state markov processes. In *Fundamentals of applied probability theory* chapter Discrete-state Markov Processes. (pp. 163–200). McGraw-Hill.
- Fu, T.-y., Lee, W.-C., & Lei, Z. (2017). Hin2vec: Explore meta-paths in heterogeneous information networks for representation learning. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management CIKM '17* (pp. 1797–1806). New York, NY, USA: ACM.
- Grover, A., & Leskovec, J. (2016). node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 855–864). ACM.
- Huang, Y., Zhu, F., Yuan, M., Deng, K., Li, Y., Ni, B., Dai, W., Yang, Q., & Zeng, J. (2015). Telco churn prediction with big data. In *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data* (pp. 607–618). ACM.
- Hughes, A. (1994). *Strategic Database Marketing: The Master Plan for Starting and Managing a Profitable, Customer-Based Marketing program*. Probus Publishing Co., Chicago, IL.
- Jiang, H., Song, Y., Wang, C., Zhang, M., & Sun, Y. (2017). Semi-supervised learning over heterogeneous information networks by ensemble of meta-graph guided random walks. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17* (pp. 1944–1950).
- Kim, K., Jun, C.-H., & Lee, J. (2014). Improved churn prediction in telecommunication industry by analyzing a large network. *Expert Systems with Applications*, 41, 6575–6584.
- McCarty, J. A., & Hastak, M. (2007). Segmentation approaches in data-mining: A comparison of rfm, chaid, and logistic regression. *Journal of business research*, 60, 656–662.
- Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013a). Efficient estimation of word representations in vector space. *CoRR*, abs/1301.3781.

- Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013b). Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems* (pp. 3111–3119).
- Mitrović, S., Baesens, B., Lemahieu, W., & De Weerd, J. (2017). Churn prediction using dynamic rfm-augmented node2vec. In R. Guidotti, A. Monreale, D. Pedreschi, & S. Abiteboul (Eds.), *Personal Analytics and Privacy. An Individual and Collective Perspective* (pp. 122–138). Cham: Springer International Publishing.
- Mitrović, S., Baesens, B., Lemahieu, W., & De Weerd, J. (2019). tcc2vec: Rfm-informed representation learning on call graphs for churn prediction. *Information Sciences*, . URL: <http://www.sciencedirect.com/science/article/pii/S0020025519301537>. doi:<https://doi.org/10.1016/j.ins.2019.02.044>.
- Modani, N., Dey, K., Gupta, R., & Godbole, S. (2013). Cdr analysis based telco churn prediction and customer behavior insights: A case study. In *International Conference on Web Information Systems Engineering* (pp. 256–269). Springer.
- Nguyen, G. H., Lee, J. B., Rossi, R. A., Ahmed, N. K., Koh, E., & Kim, S. (2018). Continuous-time dynamic network embeddings. In *Companion of the The Web Conference 2018 on The Web Conference 2018* (pp. 969–976). International World Wide Web Conferences Steering Committee.
- Perozzi, B., Al-Rfou, R., & Skiena, S. (2014). Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining* (pp. 701–710). ACM.
- Shi, Y., Gui, H., Zhu, Q., Kaplan, L. M., & Han, J. (2018). Aspem: Embedding learning by aspects in heterogeneous information networks. In *SDM* (pp. 144–152). SIAM.
- Sun, Y., & Han, J. (2013). Mining heterogeneous information networks: A structural analysis approach. *SIGKDD Explor. Newsl.*, 14, 20–28.
- Sun, Y., Han, J., Yan, X., Yu, P. S., & Wu, T. (2011). Pathsims: Meta path-based top-k similarity search in heterogeneous information networks. *Proceedings of the VLDB Endowment*, 4, 992–1003.

- Tang, J., Qu, M., & Mei, Q. (2015a). Pte: Predictive text embedding through large-scale heterogeneous text networks. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining KDD '15* (pp. 1165–1174). New York, NY, USA: ACM.
- Tang, J., Qu, M., Wang, M., Zhang, M., Yan, J., & Mei, Q. (2015b). Line: Large-scale information network embedding. In *Proceedings of the 24th International Conference on World Wide Web* (pp. 1067–1077). ACM.
- Umayaparthi, V., & Iyakutti, K. (2017). Automated feature selection and churn prediction using deep learning models. *International Research Journal of Engineering and Technology (IRJET)*, 4, 1846–1854.
- Verbeke, W., Martens, D., & Baesens, B. (2014). Social network analysis for customer churn prediction. *Applied Soft Computing*, 14, 431–446.
- Wangperawong, A., Brun, C., Laudy, O., & Pavasuthipaisit, R. (2016). Churn analysis using deep convolutional neural networks and autoencoders. *CoRR*, *abs/1604.05377*.
- Yang, C., Liu, M., Zheng, V. W., & Han, J. (2018). Meta-graph based hin spectral embedding: Methods, analyses, and insights. In *Proceedings of the 2018 IEEE International Conference on Data Mining (ICDM)* (pp. 657–666). IEEE.
- Zhang, X., Zhu, J., Xu, S., & Wan, Y. (2012). Predicting customer churn through interpersonal influence. *Knowledge-Based Systems*, 28, 97–104.

Appendix A.

Figure A.1: Results of the non-parametric pairwise DeLong, DeLong, Clarke-Pearson statistical test represented in the form of upper triangular matrix. The color bar depicts the range of p-values, whereby the values < 0.05 (in purple) denote that column method significantly outperforms the row one at the 95% confidence level. For better readability, naming pattern $\langle network\ type \rangle_ \langle meta\ path\ type \rangle$ is established with additional abbreviations, as follows: for network type: $AG_s \rightarrow s$, $AG_d \rightarrow d$, $AG_{s+ch} \rightarrow sch$, $AG_{d+ch} \rightarrow dch$, and meta path types: homogeneous $\rightarrow HOM$, local $\rightarrow LN$, alternating $\rightarrow ALT$, local-churn $\rightarrow LNCH$, alternating-churn $\rightarrow ALTCH$.

Figure A.2: Results of the non-parametric pairwise DeLong, DeLong, Clarke-Pearson statistical test represented in the form of upper triangular matrix. The color bar depicts the range of p-values, whereby the values < 0.05 (in purple) denote that column method significantly outperforms the row one at the 95% confidence level. For better readability, naming pattern $\langle network\ type \rangle_ \langle meta\ path\ type \rangle$ is established with additional abbreviations, as follows: for network type: $AG_s \rightarrow s$, $AG_d \rightarrow d$, $AG_{s+ch} \rightarrow sch$, $AG_{d+ch} \rightarrow dch$, and meta path types: homogeneous $\rightarrow HOM$, local $\rightarrow LN$, alternating $\rightarrow ALT$, local-churn $\rightarrow LNCH$, alternating-churn $\rightarrow ALTCH$.