Graph Embeddings in Practice: A Telco Churn Prediction Use Case

PhD Researcher: Sandra Mitrović

Supervisor: Prof. Dr. Jochen De Weerdt

Department of Decision Sciences and Information Management, KU Leuven

Graph Embedding Day, Lyon
07 Sept 2018

Background

Classification task

- Churn prediction (CP)
  - Predicting the probability of a customer to stop using company's services
  - Considered as the topmost challenge for Telcos [FCC report, 2009]
    - Despite **not** being novel
    - Given that acquisition costs are 5-10x higher than retention costs [Rosenberg et al, 1984]

KU LEUVEN

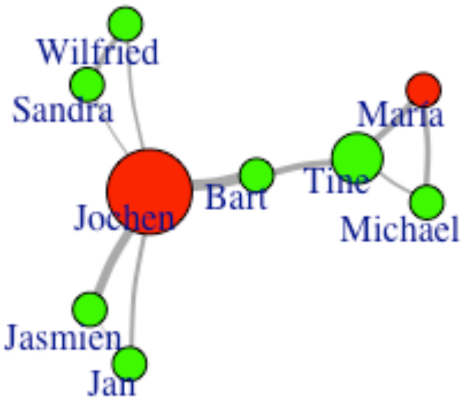
What networks have to do with CP?

- Many different data sources and approaches used

- Recently, most frequently:
    - Data source: *Usage data*
        - *Call Detail Records (CDRs)*
        - w OR w/o: Socio-demographic, Subscription, Ordering, Call center (complaints), Invoicing…
    - Approach: *Social Network Analysis (SNA)*

Date	Call Duration(sec)	Caller Number	Callee Number
2008-09-02 20:44:19	34	24002937	24997766
2008-09-02 20:42:56	26	24002937	24997766
2008-09-02 20:39:05	29	24002937	24997766
2008-09-02 20:38:06	24	24002937	24997766

- CDRs -> call graphs
    - Customer -> node
    - Call -> edge
    - Intensity of relationship -> edge weight



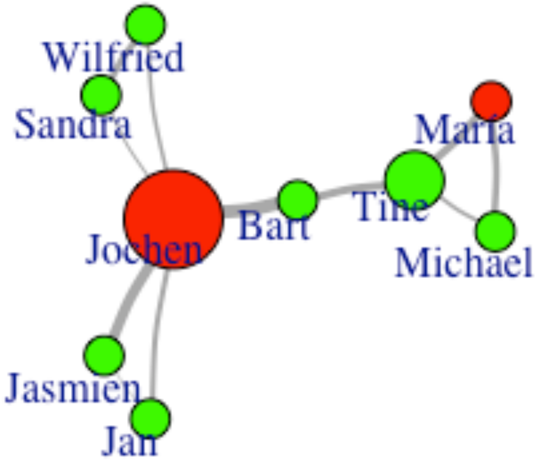
- Graph featurization

- Better predictive performance [Dasgupta et al, 2008; Richter et al, 2010; Backiel et al, 2016]

Call graph featurization

Extracting informative features from (call) graphs

- An intricate process, due to:
  - Complex structure / different types of information
    - Topology-based (structural)
    - Interaction-based (as part of customer behavior)
      - Edge weights quantifying customer behavior
  - Dynamic aspect
    - Call graph are time-evolving
    - Both nodes and edges volatile
      - Churn = lack of activity



KU LEUVEN

Shortcomings of current related work

Not many studies account for **dynamic aspects of call networks**
[Dasgupta et al, 2008; Richter et al, 2010; Kusuma et al, 2013; Huang et al, 2015; Backiel et al, 2016]

- Especially not **jointly with interaction and structural features**
  - Structural features are under-exploited [Phadke, 2013; Backiel et al, 2016]
  - Due to high computational time in large graphs (e.g. betweenness centrality) [Zhu, 2011]
- And **without using ad-hoc handcrafted features**
  - No featurization methodology [*]
  - Dataset dependent [*]

Our goal

- Performing “holistic” featurization of call graphs
  - Incorporating both interaction and structural information
  - Avoiding/reducing feature handcrafting
  - While also capturing the dynamic aspect of the network

Our goal

- Performing “holistic” featurization of call graphs
  - **Incorporating both interaction and structural information**
  - Avoiding/reducing feature handcrafting
  - While also capturing the dynamic aspect of the network

Integrating interaction and structural information

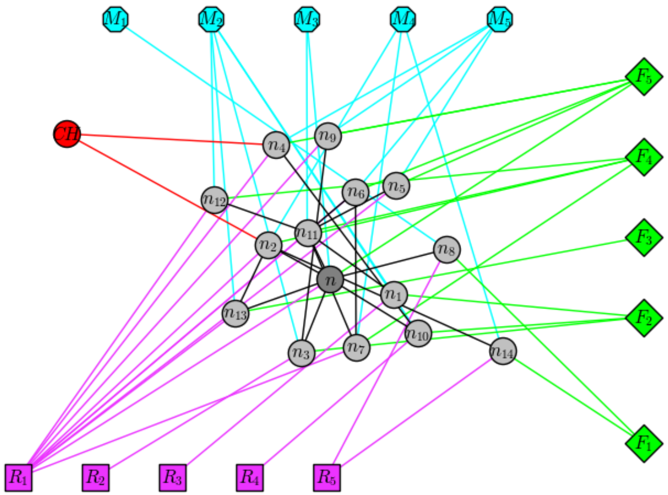
Interactions

- RFM (**R**ecency-**F**requency-**M**onetary) model [Hughes, 1994]
  - Standard for quantifying customer behavior/interactions (w.r.t. target event)
  - Many different variants found in literature
  - RFM operationalizations (our work):
    - Summary RFM (RFM_s) – total
    - Detailed RFM (RFM_d) – direction & destination sliced: $X_{out\_h}, X_{out\_o}, X_{in}$, $X \in \{R,F,M\}$
    - Churn RFM (RFM_{ch}) – only w.r.t. churners

KU LEUVEN

RFM-Augmented networks

- Original topology extended
  - By introducing artificial nodes based on RFM
  - Structural information partially preserved
- Each of R, F, M partitioned into 5 quintiles
  - One artificial node assigned to each quintile
  - Interaction info embedded through extended topology

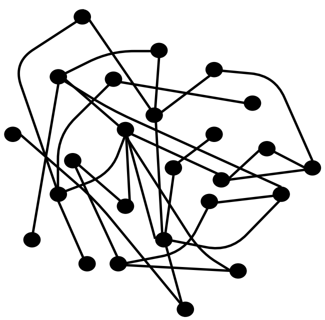


RFM features

- RFM_s
- $RFM_s \mid\mid RFM_{ch}$
- RFM_d
- $RFM_d \mid\mid RFM_{ch}$

+

Network topology



4 augmented networks

- AG_s
- AG_{s+ch}
- AG_d
- AG_{d+ch}

KU LEUVEN

Our goal

- Performing “holistic” featurization of call graphs
  - Incorporating both interaction and structural information
  - **Avoiding/reducing feature handcrafting**
  - While also capturing the dynamic aspect of the network

RL: Node2vec -> scalable node2vec

Node2vec

- Accounts both for previous and current node
- Additional parameters (p,q)
- To make walks efficient, requires precomputation of probability transitions:
  - On node level (1st time)
  - On edge level (successive)
  - Alias sampling used for efficient sampling
    - reduces O(n) to O(1)

However, does not scale well on large graphs!

(our case ~ 40M edges)

Scalable node2vec

- Accounts only for current node
- No additional parameters
- Requires precomputation of probability transitions only on node level
  - Alias sampling retained

Therefore, scales well even on large graphs!

Our goal

- Performing “holistic” featurization of call graphs
  - Incorporating both interaction and structural information
  - Avoiding/reducing feature handcrafting
  - **While also capturing the dynamic aspect of the network**

Dynamic graphs

Different definitions (current literature)

- $G = (V, E, T)$
- $G = (V, E, T, \Delta T)$
- $G = (V, E, T, \sigma, \Delta T)$

Standard approach

- Consider several static snapshots of a dynamic graph

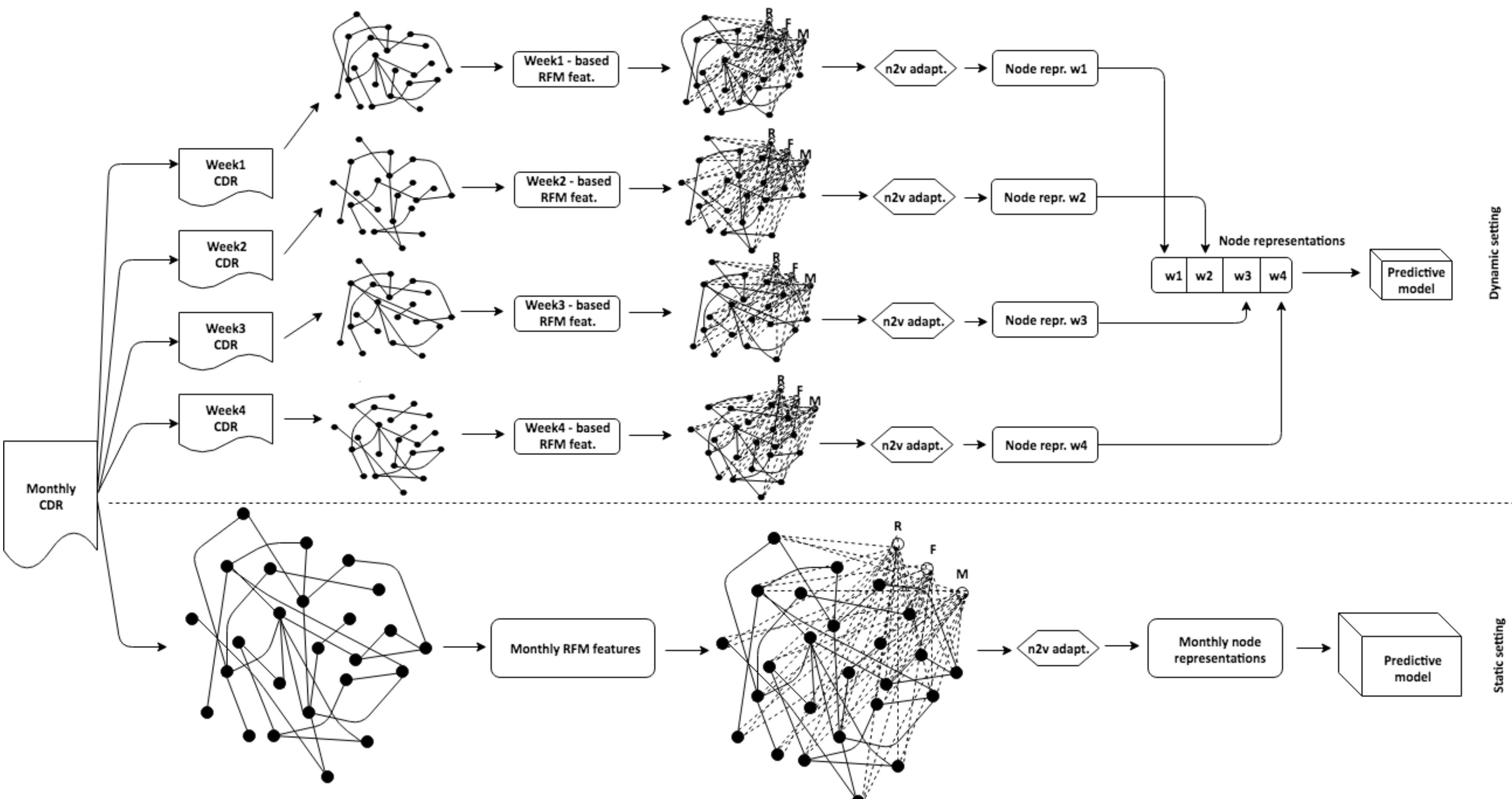
Our setting

- Monthly call graph $G = (V, E)$ ->

  Four temporal graphs $G_i = (V_i, E_i, w_i)$, $i = 1,..,4$

KU LEUVEN

Methodology – Graphical overview



KU LEUVEN

Experimental Evaluation

Research questions

- RQ1: Do features taking into account dynamic aspects perform better than static ones?
- RQ2: Do RFM-augmented network constructions improve predictive performance?
- RQ3: Does the granularity of interaction information (summary, summary +churn, detailed, detailed+churn) influence the predictive performance?

Experiments

- *RFM_s stat. vs. RFM_s dyn. vs. AG_s stat. vs. AG_s dyn. -> summary*
- *RFM_{s+ch} stat. vs. RFM_{s+ch} dyn. vs. AG_{s+ch} stat. vs. AG_{s+ch} dyn. -> summary+churn*
- *RFM_d stat. vs. RFM_d dyn. vs. AG_d stat. vs. AG_d dyn. -> detailed*
- *RFM_{d+ch} stat. vs. RFM_{d+ch} dyn. vs. AG_{d+ch} stat. vs. AG_{d+ch} dyn. -> detailed*+churn

Experimental results (1/2)

Prepaid

RFM	Static		Dynamic		Augmented network	Static		Dynamic	
	AUC	Lift	AUC	Lift		AUC	Lift	AUC	Lift
RFM_s	0.671	1.788	0.680	2.025	AG_s	0.680	**2.061**	**0.694**	2.013
RFM_{s+ch}	0.671	1.789	0.689	2.014	AG_{s+ch}	0.680	1.976	**0.705**	**2.331**
RFM_d	0.683	1.857	0.692	**2.063**	AG_d	0.678	1.898	**0.693**	2.019
RFM_{d+ch}	0.682	1.856	0.695	2.040	AG_{d+ch}	0.680	1.967	**0.702**	**2.316**

- RQ1 Answer: Dynamic better than static!
- RQ2 Answer: RFM-augmented networks improve predictive performance
- RQ3 Answer: Best performing interaction granularity is: summary+churn
  - Second best: detailed+churn

KU LEUVEN

Experimental results (2/2)

Postpaid

RFM	Static		Dynamic		Augmented network	Static		Dynamic	
	AUC	Lift	AUC	Lift		AUC	Lift	AUC	Lift
RFM_s	0.741	3.367	0.743	3.403	AG_s	0.759	3.602	**0.768**	**3.919**
RFM_{s+ch}	0.741	3.369	0.758	3.858	AG_{s+ch}	0.760	3.553	**0.769**	**3.928**
RFM_d	0.750	3.750	0.757	3.874	AG_d	0.754	3.716	**0.764**	**3.908**
RFM_{d+ch}	0.750	3.751	**0.767**	3.885	AG_{d+ch}	0.755	3.720	0.764	**3.901**

- RQ1 Answer: Dynamic better than static!
- RQ2 Answer: RFM-augmented networks improve predictive performance
- RQ3 Answer: Best performing interaction granularity is summary+churn
  - Second best: summary

KU LEUVEN

Shortcomings of current related work

- Call graphs are mostly considered to be **static** [Dasgupta et al, 2008; Richter et al, 2010; Kusuma et al, 2013; Huang et al, 2015; Backiel et al, 2016]
  - Despite: node/edge creation/deletion, node attributes/edge weights changes
  - Static approach has smoothing-out effect on customers' behavioral changes, hindering the valuable behavioral shifts leading to churn event

- Very few works explicitly address dynamic aspect
  - Time-series -based [Lee et al, 2011; Chen et al, 2012; Zhu et al, 2013]
  - Dynamic network –based (DN-based)

    DN = a series of static networks defined over non-overlapping time-intervals
    - Using **ad-hoc hand-engineered** features [Hill et al, 2006; Saravanan et al, 2012]
      - No featurization methodology
      - Featurization effort propagates through a sequence of static networks
      - Interaction and structural features underexploited
    - **No discern of difference between behavior** in different time intervals [Hill et al, 2006; Saravanan et al, 2012]

KU LEUVEN

Methodology

- We propose **sliding-window** approach
  - Overlapping intervals
  - As contrast to a single (static) and non-overlapping intervals
- We propose considering two different network types:
  - Shifted networks
  - Difference networks
- Applying RL on these networks

KU LEUVEN

Networks considered

- Shifted networks
  - Given original graph $G = (V, E)$ for the observed time period T and set of intervals $\{ [t_i, t_i+l) \}_{i=1,\dots n}$, s.t. $t_i < t_{i+1} < t_i+l$, where l is interval length
  - Shifted network $S_i = (V_i, E_i)$ corresponds to time interval $[t_i, t_i+l)$
    - **Unweighted** shifted network S^u_i (all edges equally weighted)
    - **Weighted** shifted network S^w_i

    (cum. weights of the original edges vs. artificial edges = 50:50)

- Difference networks
  - Build upon shifted networks
  - Idea: delineate differences at network level by detecting bidirectional (+/-) changes in customer activity for consecutive time intervals
  - Comparing the presence of edges and their corresponding weights (in case of a weighted graph)

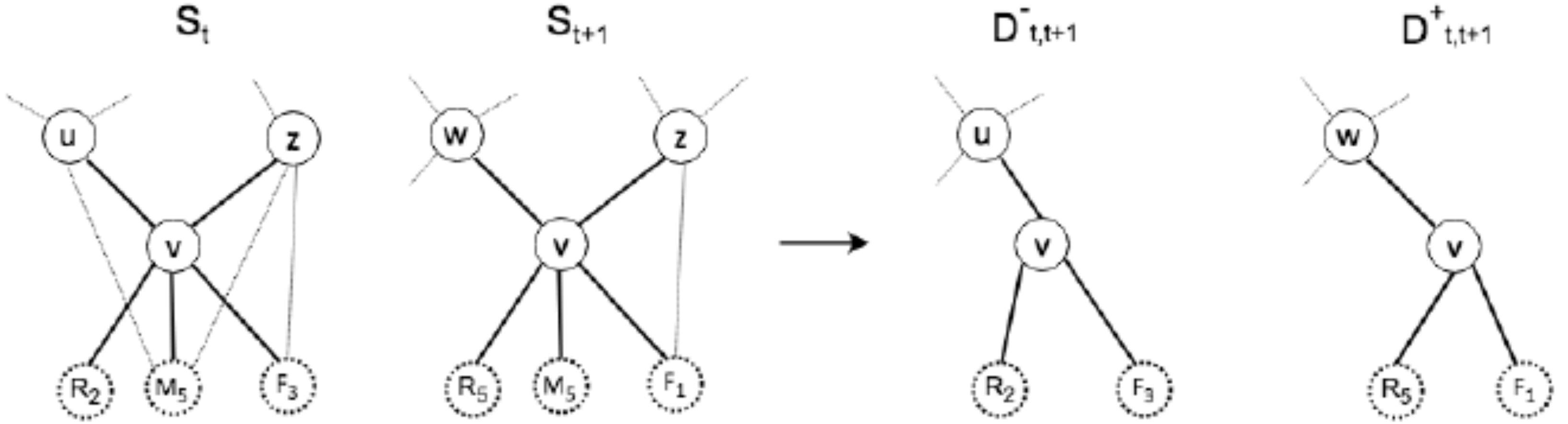
Derivation of difference networks (1/2)

Original network (UW) / Unweighted artificial (UWA)

- Given shifted networks $S_i = (V_i, E_i)$ and $S_j = (V_j, E_j)$ where $t_i < t_j$:
  - Decreased difference network

    $D_{ij}^- = (V_{ij}^-, E_{ij}^-)$ with $E_{ij}^- = \{e \text{ with weight } w_e^i, \text{ if } e \in E_i \setminus E_j\} \cup \{e \text{ with weight } |w_e^j - w_e^i|, \text{ if } e \in E_i \cap E_j \text{ and } w_e^j - w_e^i < 0\}$
  - Increased difference network

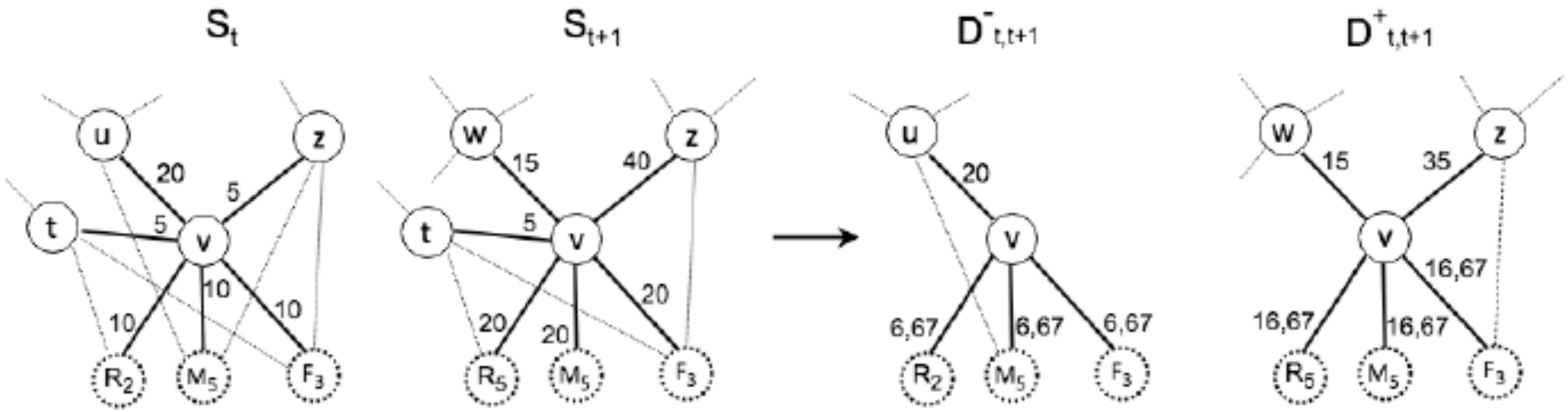
    $D_{ij}^+ = (V_{ij}^+, E_{ij}^+)$ with $E_{ij}^+ = \{e \text{ with weight } w_e^i, \text{ if } e \in E_j \setminus E_i\} \cup \{e \text{ with weight } w_e^j - w_e^i, \text{ if } e \in E_i \cap E_j \text{ and } w_e^j - w_e^i > 0\}$



Derivation of difference networks (2/2)

Weighted network (W)

- First: consider artificial edges as unweighted in order to detect differences in edges (previous case)
- Next: for the remaining ones we perform weights scaling to maintain the ratio between cumulative weights (original edges vs. artificial edges) be 50:50.



KU LEUVEN

Experimental Evaluation

Setting:

- Two datasets – one prepaid, one postpaid
- Nine overlapping time intervals considered
- Stacked representations input to l2-regularized logistic regression
- Evaluation in terms of AUC & lift

Goal:

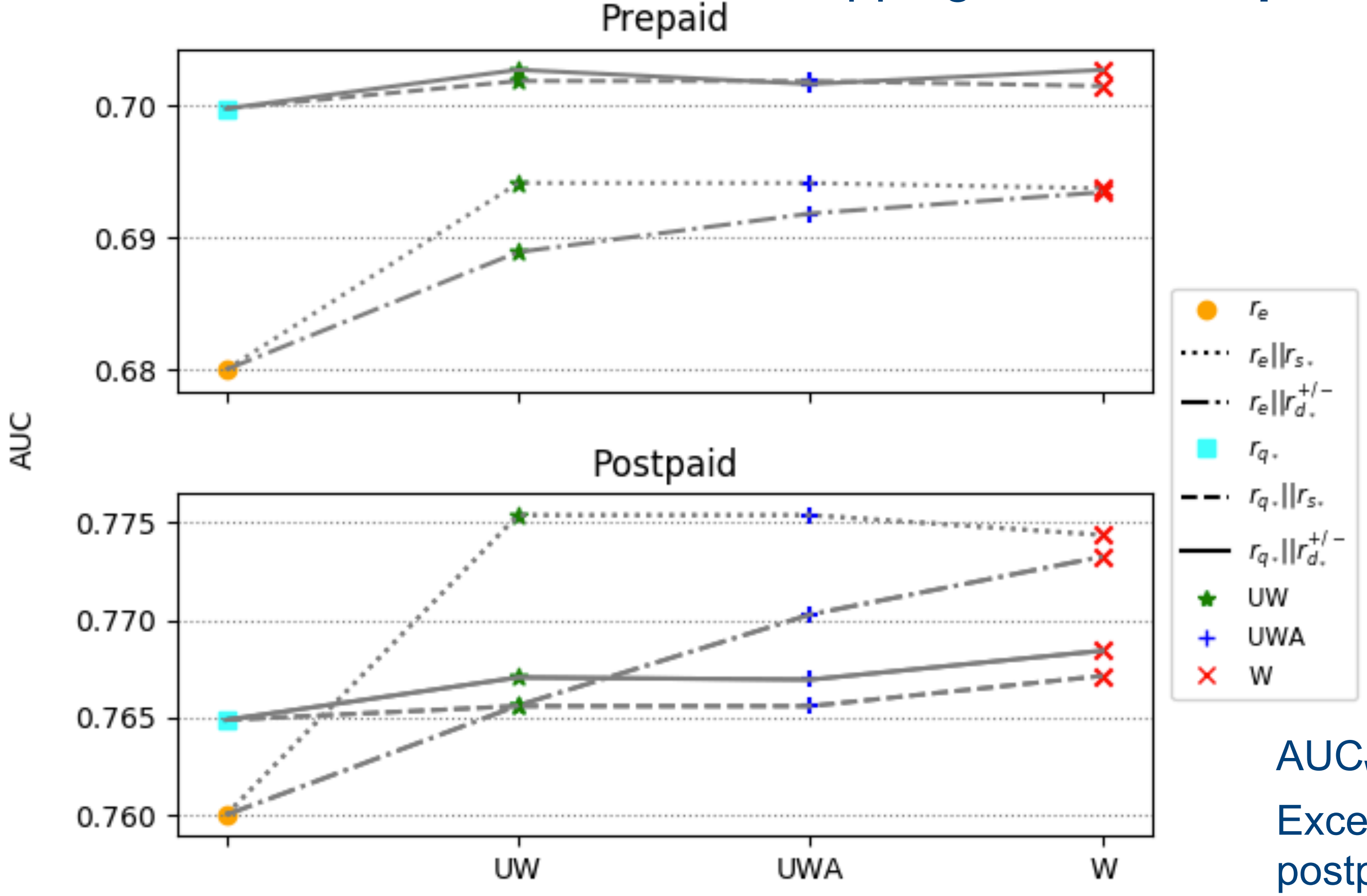
- Compare predictive performance of different representations obtained on various time periods (and corresponding networks)

Notation	Definition
$r_e(v)$	Node v repr. obtained on the **entire-period** network
$r_{q_i}(v)$	Node v repr. obtained on **quarter-of-period** network w_i
$r_{s_i}(v)$	Node v repr. obtained on **shifted** network S_i
$\Delta r_{s_{ij}}(v)$	Vector difference of node v repr. obtained on two consecutive shifted networks S_i and S_j
$r^+_{d_{ij}}(v)$	Node v repr. obtained on **increase difference** network D^+_{ij}
$r^-_{d_{ij}}(v)$	Node v repr. obtained on **decrease difference** network D^-_{ij}

KU LEUVEN

Experimental Results

- Adding shifted and difference network –based representations to static and the one based on non-overlapping intervals **improves** AUC



$AUC_W > AUC_{UW/UWA}$

Except for r_e || r_{s*} for postpaid

Experimental Results

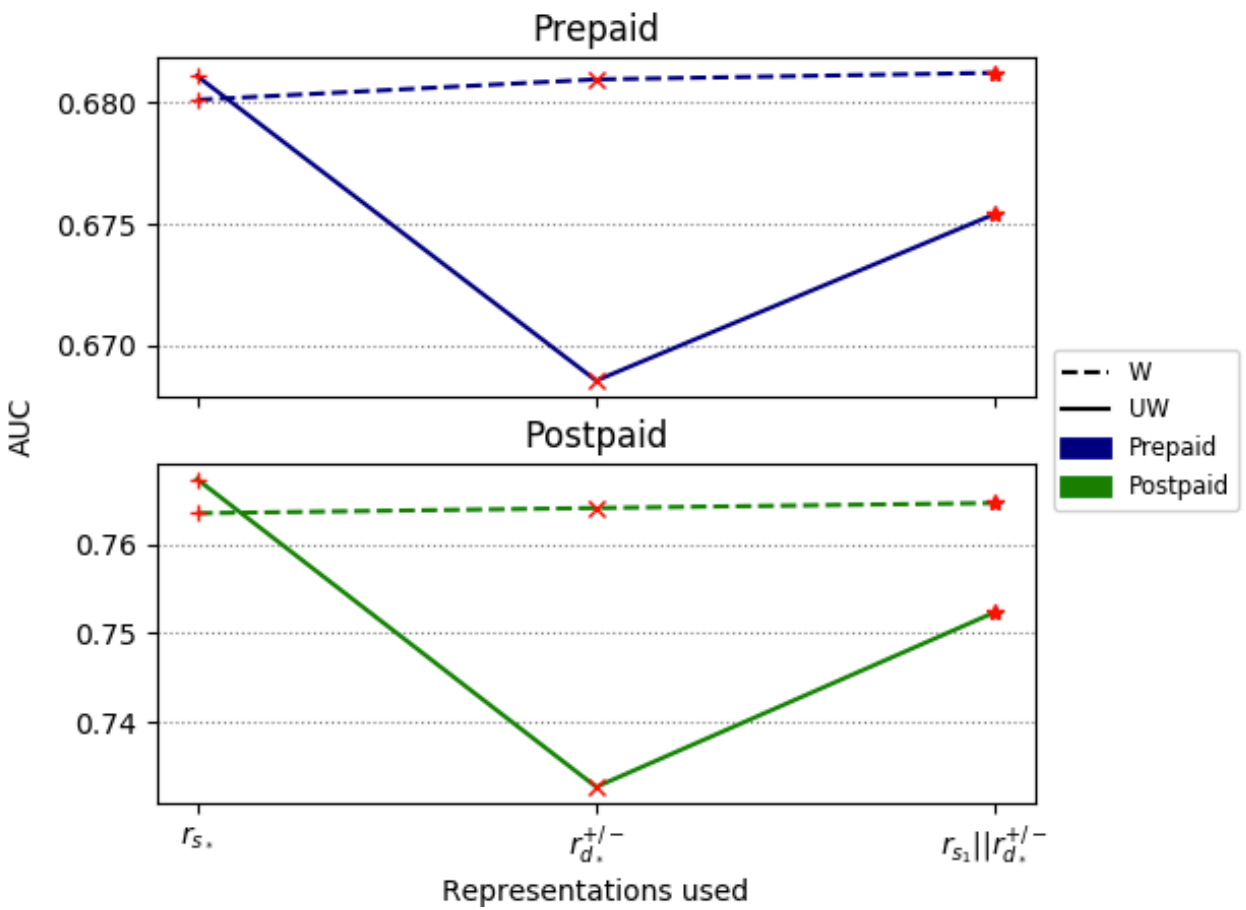
Dataset	r_e	r_{q_*}	Type	Shifted			Delta			Difference		
				r_{s_*}	$r_{s_*} \| r_e$	$r_{s_*} \| r_{q_*}$	Δr_{s_*}	$\Delta r_{s_*} \| r_e$	$\Delta r_{s_*} \| r_{q_*}$	$r_{d_*}^{+/-}$	$r_{d_*}^{+/-} \| r_e$	$r_{d_*}^{+/-} \| r_{q_*}$
Prepaid	0.68000 (1.97600)	0.69978 (2.36861)	W	0.68010 (1.90333)	0.69374 (2.08470)	0.70149 (2.28820)	0.67441 (1.82053)	0.69236 (2.06782)	0.70142 (2.28422)	0.68094 (1.89887)	0.69344 (2.03709)	**0.70271** (2.29457)
			UW	0.68108 (1.92785)	0.69414 (2.07769)	0.70187 (2.29345)	0.67373 (1.80206)	0.69210 (2.06384)	0.70120 (2.28151)	0.66856 (1.78422)	0.68891 (1.96129)	<u>**0.70272**</u> (2.29154)
			UWA							0.67881 (1.94855)	0.69183 (2.06081)	0.70164 (2.29218)
Postpaid	0.76000 (3.55300)	0.76488 (4.10355)	W	0.76346 (3.92656)	**0.77437** (3.82203)	0.76714 (3.94654)	0.75490 (3.78977)	0.77072 (3.78158)	0.76597 (3.91716)	0.76405 (3.94654)	0.77326 (3.83070)	0.76843 (3.94437)
			UW	0.76729 (3.95400)	<u>**0.77539**</u> (3.83143)	0.76559 (3.89982)	0.76072 (3.81120)	0.77230 (3.78350)	0.76687 (3.90849)	0.73271 (3.50054)	0.76562 (3.73462)	0.76706 (3.93570)
			UWA							0.75976 (3.89091)	0.77029 (3.81337)	0.76695 (3.92318)

- Comparing r_e, r_{q*}, r_{s*}, $r_{d*}^{+/-}$ (in terms of AUC):
  - r_{q*} outperforms others except for postpaid unweigthed (r_{s*})
  - Weighted: r_e performs the worst
  - Unweighted: $r_{d*}^{+/-}$ performs the worst
- Comparing shifted and difference (in terms of AUC):
  - Weighted: $r_{d*}^{+/-}$ outperforms r_{s*}
  - Unweighted: r_{s*} outperforms $r_{d*}^{+/-}$
  - Combining r_{s*} and $r_{d*}^{+/-}$ with r_e, r_{q*} results become dataset-dependent

KU LEUVEN

Additional analysis

- $r_{s1} \parallel r_{d*}^{+/-}$



- The results improved, but still could not win r_{s*} for unweighted

Conclusion

- We designed **RFM-augmentations** of original graphs
  - Enable conjoining interaction and structural information
- We devise a **scalable** adaption of the original node2vec approach
  - Relaxing random walk generation and avoiding grid search tuning for two additional parameters
- We attempt to take into account dynamic aspect of the networks
  - We propose applying **representation learning on top of:**
    - Networks obtained from non-overlapping intervals
    - Shifted networks (overlapping intervals)
    - Difference networks

  to **explicitly capture changes** in customer behavior.
- We demonstrate that compared to only static, non-overlapping intervals-based dynamic representations perform better and **adding shifted/difference** network representations **results in even better performance improvements.**

KU LEUVEN

Future research

- Experiment with more sophisticated methods for assessing dynamic differences in customer behavior
- Analyzing the effect of applying temporal random walks
- Investigating how different approaches which involve shifting temporal aspect into the RL part affect predictive performance

References

- FCC, 2009. *13th Annual report and analysis of competitive market conditions with respect to mobile wireless, including commercial mobile services*, Federal Communication Commission, WT Docket 10-133.
- Verbeke et al., 2010. *Customer churn prediction: does technique matter?* In Proceedings of the Joint Statistical Meeting, JSM2010, Vancouver, Canada.
- Grover and Leskovec*, 2016. Node2Vec: Scalable Feature Learning for Networks.* In Proceedings of KDD '16, San Fransicso, California, US.
- Mikolov et al., 2013. *Distributed representations of words and phrases and their compositionality.* In Advances in neural information processing systems (pp. 3111-3119).
- Perozzi et al., 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.
- Tang et al., 2015. *Line: Large-scale information network embedding.* In Proceedings of the 24th International Conference on World Wide Web (pp. 1067-1077). ACM. Chicago.

- Grover and Leskovec*, 2016. Node2Vec: Scalable Feature Learning for Networks.* In *Proceedings of KDD '16, San Fransicso, California, US.*

Bibliography

- Mitrovic et al., 2017a. *Scalable RFM-enriched Representation Learning for Churn Prediction. DSAA 2017: 79-88.*
- Mitrovic et al., 2017b. *Churn Prediction Using Dynamic RFM-Augmented Node2vec. PAP@PKDD/ECML 2017: 122-138.*
- Mitrovic et al., 2018. *Dyn2Vec: Exploiting dynamic behaviour using difference networks-based node embeddings for classification. ICDATA 2018: 194-200.*

Thank you!

Questions?

Email: sandra.mitrovic@kuleuven.be