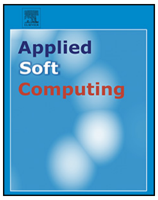
Applied Soft Computing 14 (2014) 431–446

Contents lists available at ScienceDirect

Applied Soft Computing

journal homepage: www.elsevier.com/locate/asoc



Social network analysis for customer churn prediction

Wouter Verbeke[a,*], David Martens[b], Bart Baesens[c,d,e]

[a] Vrije Universiteit Brussel, Faculty of Economic, Political and Social Sciences and Solvay Business School, Pleinlaan 2, B-1050 Brussels, Belgium
[b] University of Antwerp, Faculty of Applied Economics, Prinsstraat 13, B-2000 Antwerp, Belgium
[c] Katholieke Universiteit Leuven, Department of Decision Sciences and Information Management, Naamsestraat 69, B-3000 Leuven, Belgium
[d] University of Southampton, School of Management, Highfield Southampton SO17 1BJ, United Kingdom
[e] Vlerick, Leuven-Ghent Management School, Reep 1, B-9000 Ghent, Belgium

ARTICLE INFO

Article history:
Received 27 December 2012
Received in revised form 18 May 2013
Accepted 17 September 2013
Available online 16 October 2013

Keywords:
Data mining
Social networks
Customer churn prediction
Relational classification
CDR data
Profit

ABSTRACT

This study examines the use of social network information for customer churn prediction. An alternative modeling approach using relational learning algorithms is developed to incorporate social network effects within a customer churn prediction setting, in order to handle large scale networks, a time dependent class label, and a skewed class distribution. An innovative approach to incorporate non-Markovian network effects within relational classifiers and a novel parallel modeling setup to combine a relational and non-relational classification model are introduced. The results of two real life case studies on large scale telco data sets are presented, containing both networked (call detail records) and non-networked (customer related) information about millions of subscribers. A significant impact of social network effects, including non-Markovian effects, on the performance of a customer churn prediction model is found, and the parallel model setup is shown to boost the profits generated by a retention campaign.

1. Introduction

In recent years, vast amounts of networked data on a broad range of network processes and information flows between interlinked entities have become available, such as calls and text messages linking telephone accounts [12] and money transfers connecting bank accounts [26]. These massive, networked data logs potentially hide information that is highly valuable to companies and organizations, and as such open new perspectives for innovative business applications [4,38].

Networked data present both complications and opportunities for predictive data mining. The data are patently not independent and identically distributed, which introduces bias to learning and inference procedures [19,24]. Relational learning aims to exploit the information contained within the network structure of data instances, and to incorporate this information within a network classification or regression model [13,16]. Relational classifiers (RC) learn directly from a graph or network, as opposed to non-relational classifiers (N-RC) which require an attribute-value representation of the data.

The central research question in this study consists of a *theoretical* and an *applied* component, concerning respectively the *use* and the *merits* of social network information for customer churn prediction in the telco industry. More specifically, this study develops a number of approaches to mine social network information for churn prediction, and examines the effect on the predictive power of the resulting models. Customer churn undermines the profitability of telco operators, facing annual churn rates up to 20% and higher. Therefore customer relationship management, and more specifically customer retention, receives a growing amount of attention from telco operators. Customer churn prediction (CCP) models aim to detect customers with a high propensity to attrite. This allows a company to improve the efficiency of retention campaigns which aim to prevent customers from churning, by directing personalized retention efforts to the customers that are effectively about to churn.

Fig. 1 introduces a general framework for customer churn prediction modeling in the telco industry. The top panel of the figure describes the current state-of-the-art approaches. A telco operator has access to two main data sources[1]: (1) subscription records, i.e. the information a customer provides when subscribing to a service (e.g. postal address, name, etc.); (2) call detail records (CDRs), i.e.

* Corresponding author. Tel.: +32 2 629 20 53.
E-mail addresses: wouter.verbeke@vub.ac.be (W. Verbeke), david.martens@ua.ac.be (D. Martens), bart.baesens@econ.kuleuven.be (B. Baesens).

[1] A third source concerns external data providers, which have not been included to keep the framework transparent.

1568-4946/$ – see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.asoc.2013.09.017

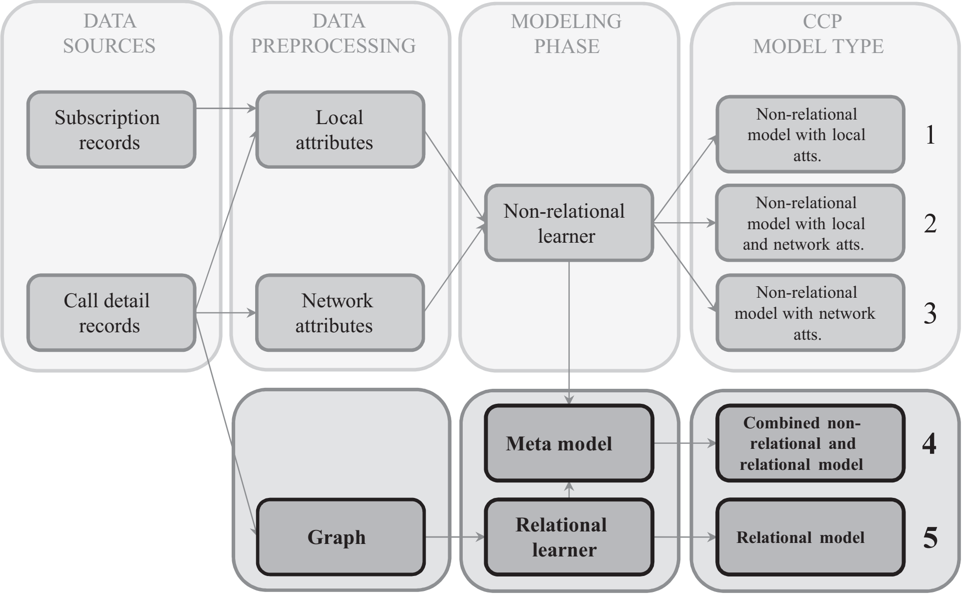


Fig. 1. General framework for customer churn prediction in the telco industry, with the current modeling approaches depicted in the top panel and the approaches developed in this study in the bottom panel.

communication logs describing the identity (i.e. the phone number) and operator of interacting subscribers, and the exact type, time, and duration of each interaction. Typically, subscription records are only available for the contractual postpaid (as opposed to the non-contractual prepaid) customer segment. These records can be converted into local attributes, which summarize general information related to individual customers, and network attributes, which aggregate information related to the (social) network of a customer. Using historical data, non-relational classifiers are able to learn a classification model to predict future churn based on these attributes. A differentiation between the type of CCP model is made based on the types of attributes that are included in the model. Currently, a great focus exists in the industry on the use of network attributes in order to incorporate social network effects and to improve the predictive power of a CCP model (yielding models of types 2 and 3 in Fig. 1). However, to our knowledge, the merits of incorporating social network attributes in a CCP model thus far have not been studied thoroughly in the literature. Therefore, in this study the impact of network attributes on the predictive power of a non-relational CCP model will be examined in more detail.

As an alternative to the currently applied non-relational approaches discussed above, this study develops an alternative approach which is depicted in the bottom panel of Fig. 1. CDRs can be converted into a graph, with the nodes in the graph representing subscribers and the links between the nodes representing social ties between the subscribers. This graph is called the call graph [30] and represents the social network of the subscribers of the telco operator. From a theoretical perspective, this paper contributes by developing relational classifiers that allow to incorporate social network effects within a CCP model and to handle the specific characteristics of a customer churn prediction setting. For instance, the class distribution of churners and non-churners is typically very skewed, since there are much less churners than non-churners. This may cause existing data mining techniques to experience difficulties in learning powerful models. In a relational learning context, typical methods to handle a skewed class distribution, such as sampling techniques, are not applicable. Therefore, the existence of non-Markovian or higher order social network effects and their use for customer churn prediction is examined. Relational learners typically restrict the impact of the network on a node to the first order neighborhood, i.e. the nodes in the network that are directly connected to a particular node (e.g. Macskassy and Provost [24], Neville and Jensen [32]). Existing relational learners are extended in order to incorporate higher order network effects by allowing them to take into account higher order network effects *as if* first order network effects. Furthermore, existing techniques are adjusted to properly handle the time dimension present in customer churn, as well as to deal with the massive size of the graph representing the social network of the customer base of a telco operator, which typically consists of millions of subscribers. Finally, a number of approaches to combine a non-relational and a relational CCP model are presented, aiming to reinforce the predictive power of current CCP modeling approaches by adding a relational model. From an application perspective, this study contributes by evaluating and comparing the CCP modeling approaches summarized in Fig. 1 in two extensive, real life case studies. This will allow to assess the impact of social network attributes on the performance of a non-relational CCP model, as well as to compare a non-relational, a relational, and a combined CCP model. The two case studies concern the prediction of churn in respectively a prepaid and a postpaid customer segment. To this end, two large-scale, real life data sets have been obtained containing networked (CDRs) and non-networked (usage statistics, sociodemographic, marketing related) information about millions of customers. The results of the experiments will be assessed using lift as well as the novel maximum profit measure, which allows to assess the performance of a CCP model from a profit point of view.

2. Social network information for customer churn prediction

2.1. Graph theoretical definitions and notations

Boccaletti et al. [3] defines graph theory as the natural framework for the exact mathematical treatment of complex networks, and formally, a complex network can be represented as a graph. In this study the graph and the complex network coincide, since the analyzed networks are approximate mathematical representations of the social ties between the customers of a telco operator. A graph $\mathbf{G}$ consists of a set of vertices (or nodes, or points) $v \in \mathbf{V}$

that are connected by a set of edges (or links, or lines) $e \in \mathbf{E}$, and $\mathbf{G} = (\mathbf{V}, \mathbf{E})$. The number of elements in the sets **V** and **E** are denoted respectively by n and k. The links in a graph can be *directed* or *undirected*. Directed links point from an origin to a destination node and incorporate a direction property, while undirected links do not. In an undirected graph maximum one link exists between two nodes v_i and v_j, whereas in a directed graph maximum two links exist between two customers v_i and v_j, with one link representing the interactions going from v_i towards v_j and the other representing the interactions from v_j towards v_i.

A graph can be represented by an adjacency matrix $\mathbf{A} = (a_{ij})$ or a weight matrix $\mathbf{W} = (w_{ij})$ of size $n \times n$. An entry a_{ij} or w_{ij} represents the edge between vertex v_i and vertex v_j. The value of a_{ij} in an adjacency matrix is equal to one if an edge exists between vertices v_i and v_j, and equal to zero when no connection exists. In principle, the weights $w_{ij} \in \mathbf{W}$ can take any value. Typically, a weight expresses a characteristic of a link, such as the strength of a social tie between two customers of a telco operator in a call graph. A value of w_{ij} equal to zero means that no relation exists between two vertices. When the weights represent distances, e.g. the length of links in traffic networks, the weight matrix is called the *distance matrix*. The values a_{ii} on the diagonal of an adjacency matrix depend on the convention that is adopted, and either equal once or twice the number of edges from vertex v_i to itself or so-called loops. In case of a weight matrix, the values w_{ii} on the diagonal depend on the property expressed by the weights. For instance, when the weights represent distances, the values on the diagonal will be equal to zero. In this study, the values of a_{ii} and w_{ii} will be set equal to zero by definition.

Defintion 1. The order o network neighborhood $\mathcal{N}_i^o$ of node v_i is defined as the subset of nodes of the set of all nodes in the network **V** that are directly or indirectly connected to node v_i, with the maximum number of links constituting the shortest path between the nodes in the neighborhood and the node v_i equal to the order o, and including node v_i itself.

The first order neighborhood $\mathcal{N}_i^1$ or equivalent $\mathcal{N}_i$ of v_i consists of all the nodes in the network that are directly connected to v_i and node v_i itself. The second order neighborhood adds to the first order neighborhood the nodes that are directly connected to the nodes in the first order neighborhood, which are not already in the first order neighborhood, etc. The neighborhood $\mathcal{N}_i^0$ of order zero is a singleton with element node v_i. Section 5 will discuss some further properties of graphs. For an extensive overview on graph theory, one may refer to, e.g. Newman [33].

2.2. Related work

A limited number of related prior studies have proposed approaches to use social network information in order to predict customer churn. Nanavati et al. [30] analyzed the structure and evolution of a massive telecommunication or call graph for a single mobile operator for four different regions in India with different socio-demographic, urbanization, and cultural characteristics, and with the number of nodes for the regions ranging up to 1.25 million. A weight matrix is constructed which only contains information about intra-regional calls, i.e. intra network calls. The period of data gathering was also different for the four regions, ranging from a week to a month.

Dasgupta et al. [12] presumably analyze the same network as Nanavati et al. [30], and is to our knowledge the first of its kind in predicting customer churn using social ties between the subscribers of a telco operator. The study focuses on the prepaid segment of customers, for which CDR data are indicated to be the only available source of information. A diffusion based modeling technique to predict customer churn is developed, which will be discussed in detail in Section 3 and applied in the case studies reported in Section 5. As an interesting topic for future research, Dasgupta et al. [12] indicate the possibility to apply relational classifiers to predict customer churn using CDR data, and specifically NetKit, as developed by Macskassy and Provost [24] and implemented and applied in this study. The results of the case study reported by Dasgupta et al. [12] confirm the existence and relevance of social network information for customer churn prediction. However, one of the major differences with this study concerns the churn rate, which is much smaller in the data sets that are used in the case studies in this paper. A skewed class distribution causes relational learners and classification techniques in general to experience difficulties in learning powerful classification models.

Recently, Richter et al. [36] presented the group-first churn prediction approach to predict customer churn based on the analysis of social groups or communities derived from CDR data. The presented approach assigns a churn score to each subscriber based on the churn score of the social group as well as personal characteristics. The results of the study reconfirm the potential of improving the current generation of CCP models by adding information that captures the social interactions of a subscriber, and indicates that group structure and membership are determinants of churn behavior. This study opens interesting alternative modeling approaches to exploit the information contained within the network structure of the customers of a telco operator. However, given the essential differences with the relational learners that are applied in this study, the group-first approach has not been included in the experiments. Numerous non-relational classification techniques have been adopted for churn prediction and customer targeting in general, including traditional statistical methods such as logistic regression [22,31,8] and Bayesian techniques, non-parametric statistical models like for instance k-nearest neighbor, decision trees, and neural networks [1,7]. An extensive overview and discussion of the literature on customer churn prediction can be found in Verbeke et al. [40], and a large scale benchmarking study comparing the performance of a range of non-relational classification techniques for customer churn prediction can be found in Verbeke et al. [39]. For a broad discussion on multivariate dependencies and a general work on the link between statistical pattern recognition and graph analysis, one may refer to respectively Cox and Wermuth [11] and Marchette [25], which provide an extensive overview and full elaboration of these topics. A recent related publication on the application of algebraic geometry to graphs has been published by Johannsen and Marchette [20].

Explaining the classifications of network-based models is an important issue for the practical use and acceptance of such models (see e.g. Gregor and Benbasat [17]); Martens and Provost [27]) and makes for an interesting topic for future research. Rule extraction techniques [1,28] could be applied to the propositional representations that provide a global explanation of the prediction model. Recently also instance-based approaches have been proposed that explain the prediction of a single data instance; a relevant approach that works on high-dimensional data (originally on textual data) that could be applied in this setting is that of Martens and Provost [27].

2.3. Evaluating customer churn prediction models

CCP models are typically evaluated using lift. The lift curve plots the churn rate among the top fraction of customers with the highest predicted probabilities to churn on the x-axis, divided by the overall base churn rate in the entire customer base. Lift indicates how much better a model identifies churners compared to randomly targeting a fraction of customers, and as such provides an intuitive measure of model performance. However, the lift curves of different models may intersect, and the highest lift is obtained by a different model

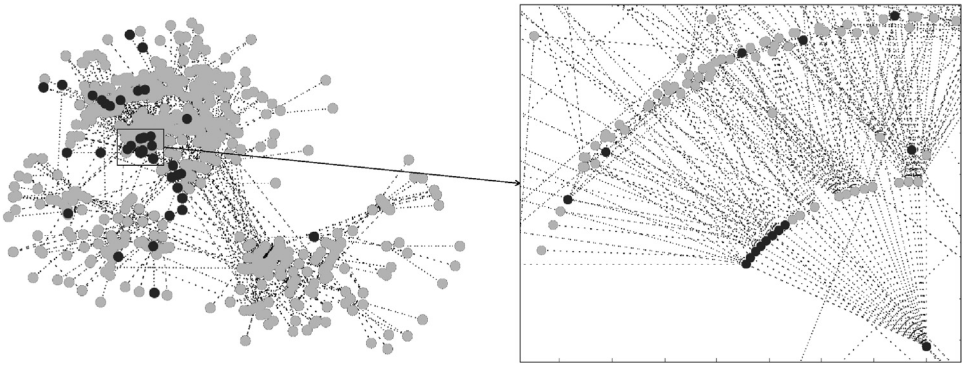


Fig. 2. The neighborhood of order eight of a particular customer in the call graph, with churners represented by black dots, and non-churners represented by gray dots. A sequence of twelve subsequent churners can be found in this network neighborhood, indicating a viral-like propagation or spreading of churn throughout the call graph.

depending on the top fraction that is selected. Therefore, using lift curves to compare the performance of CCP models may not provide a conclusive answer as to which model performs best. Moreover, the commonly used top-decile lift may lead to suboptimal model selection as well, since setting the value of the top fraction to ten percent is arbitrary.

Therefore, a performance measure to evaluate CCP models from a profit centric point of view will be applied that was recently introduced by Verbeke et al. [39]. This performance measure builds on an expression introduced by Neslin et al. [31] to calculate the profit associated with a customer retention campaign that targets subscribers based on the outcomes of a CCP model.

Defintion 2. The total profit P_t generated by a retention campaign equals:

$$P_t = n\eta\left[\left(\gamma CLV + \delta(1-\gamma)\right)\pi_0\lambda - \delta - \phi\right] - A, \tag{1}$$

with η the fraction of the customer base that is targeted, CLV the average customer lifetime value, δ the cost of the incentive, ϕ the cost of contacting the customer, and A the fixed administrative costs. The lift coefficient, λ, is the percentage of churners within the targeted fraction η of customers, divided by the base churn rate, π_0. Finally, γ is the probability of a targeted churner to be retained by the offered incentive. It is assumed that all parameters are positive, and that $CLV > \delta$.

Eq. (1) states that the profit resulting from a retention campaign to prevent customers from churning by offering an incentive to remain loyal, equals the saved value (CLV) associated with retained churners (fraction γ of the churners included in the campaign) minus the costs of the campaign. The costs of the campaign equal (1) the cost of the accepted incentives (δ) by the non-churners and the fraction of churners that are retained, (2) the cost of contacting the fraction of the customers included in the retention campaign (ϕ), and (3) a fixed administrative cost (A).

The total profit generated by a retention campaign hence depends on the fraction of included customers η, and the lift associated with this fraction, which is directly associated with the CCP model. Hence, the included fraction of customers needs to be optimized in order to maximize the generated profits by a retention campaign.

Defintion 3. The maximum profit measure for customer churn (MPC) is defined as the maximum profit that can be obtained by using the outcomes of a customer churn prediction model, and which can be used to evaluate the predictive power of a CCP model:

$$\text{MPC} = \max_{\forall\eta} P_t(\eta; \gamma, CLV, \delta, \phi). \tag{2}$$

Customer retention campaigns aim to minimize the costs associated with customer churn, and therefore it is straightforward to evaluate and select a CCP model by using the maximum profit that can be generated as a performance measure. In this study, both the lift and the MPC will be applied to assess the performance of the generated models, providing complementary insights.

3. Classification in networked data

The basic premise for customer churn prediction using social network information is that customers interlinked with customers that have churned are more likely to churn themselves. Possible explanations for the existence of such *network effects* are the word-of-mouth effect, social leader influence, promotional offers from operators to acquire groups of friends, and reduced tariffs for intra-operator traffic. The principle that a contact between similar people occurs at a higher rate than among dissimilar people has been observed in many kinds of social networks and is called homophily [2,29,24], assortativity [33], or relational autocorrelation [19].

An indication of the existence of network effects on the churn behavior of telco subscribers is provided in Fig. 2, which plots the network neighborhood of a particular customer in the CDR data set presented in Section 5 up to degree eight. A viral like spreading of churn can be observed in this part of the call graph, and a churn rate that is substantially higher than the base churn rate in the entire call graph. Many of such network effects can be observed, hinting towards a great potential of exploiting social network information for churn prediction.

The following subsections discuss the presented approaches in the framework of Fig. 1 to incorporate social network information within a customer churn prediction model. The first two subsections examine the use of relational learning techniques, while the third subsection explores the use of network variables in a non-relational model. The fourth subsection finally presents a range of meta modeling approaches to combine a relational and a non-relational model.

3.1. Relational learning for customer churn prediction

Macskassy and Provost [24] introduced a framework for learning in networked data. In this node-centric framework, a relational learner comprises a relational classifier and optionally a collective inference (CI) procedure. The original formulation of relational learners as described in Macskassy and Provost [24] assumes that part of the class labels of the nodes in the network are known, and can be used to estimate the unknown class labels. In other words, the set of vertices $\mathbf{V}$ consists of a subset of vertices with a known label, $\mathbf{V}^K$, and a subset of vertices with an unknown label, $\mathbf{V}^U$, i.e.

Table 1
Summary of relational classifiers.

RC	Summary
CDRN	The *Class-Distribution Relational Neighbor classifier* [37,34,35] learns a model based on the distribution of neighbor class labels. The class vector $CV(v_i)$ of a node v_i is defined as the vector of summed linkage weights to the various classes: $CV(v_i)_k = \sum_{v_j \in \mathcal{N}_i} w_{ij} \cdot P(l_j = c_k \vert \mathcal{N}_j)$ (3) with l_j the non-specified label of node v_j. The reference vector $RV(c)$ of a class c is defined as the average of the class vectors for nodes known to be of class c: $RV(c) = \frac{1}{\vert \mathbf{V}_c^K \vert} \sum_{v_i \in \mathbf{V}_c^K} CV(v_i)$ (4) with $\mathbf{V}_c^K = \{v_i \vert v_i \in \mathbf{V}^K, l_i = c\}$, and $\mathbf{V}^K \subset V$ the subset of vertices with a known class label. Then the probability for a customer to be a churner can be calculated as the normalized vector similarity between v_i's class vector and the churners' class reference vector: $P(l_i = c \vert \mathcal{N}_i) = sim(CV(v_i), RV(c))$(5) where $sim(a, b)$ can be any vector similarity measure normalized to lie in the range [0, 1]. In the experimental section of this study cosine similarity will be applied.
NLB	The *Network-only Link Based classifier* [23] applies logistic regression on feature vectors which are constructed for each node by aggregating the labels of neighboring nodes, e.g. existence (binary), the mode, and value counts. The count model is equivalent to the class vector $CV(v_i)$ defined in Eq. (3), and has been shown to perform best by Lu and Getoor [23]. $CV(v_i)_k = \frac{\sum_{v_j \in \mathcal{N}_i} w_{ij} \cdot P(l_j = c_k \vert \mathcal{N}_j)}{\sum_{v_j \in \mathcal{N}_i} w_{ij}}$ (6) $P(l_i = c \vert \mathcal{N}_i) = \frac{1}{1 + e^{-\beta_0 - \boldsymbol{\beta} \cdot CV(v_i)}}$ (7) with β_0 and β representing the parameters of the logistic regression model.
SPA RC	The *Spreading Activation Relational Classifier* is based on the Spreading Activation technique (SPA) proposed by Dasgupta et al. [12]. The original SPA technique models the propagation of churn through a network as a diffusion process of *churn energy*. Incorporating SPA within the modular framework of Macskassy and Provost [24] results in the SPA RC relational classifier, defined as follows: $P(l_i = c \vert \mathcal{N}_i) = \frac{1}{Z} \sum_{v_j \in \mathcal{N}_i} d \cdot \frac{w_{ji}}{\sum_{s \in \mathcal{N}_j} w_{js}} \cdot P(l_j = c \vert \mathcal{N}_j)$ (8) where Z is a normalizer to convert the energy levels in probability scores, and d a parameter that controls the diffusion process. This expression is similar to the WVRN classifier, but now the impact of a neighboring node v_j on node v_i does not depend on the relative weight w_{ij} within the neighborhood of node v_i, i.e. $\mathcal{N}_i$, but instead on the relative weight within the neighborhood of node v_j, i.e. $\mathcal{N}_j$.
WVRN	The *Weighted-Vote Relational Neighbor classifier* estimates the probability of a customer to churn as a function of the probabilities of its neighbors to churn: $P(l_i = c \vert \mathcal{N}_i) = \frac{1}{Z} \sum_{v_j \in \mathcal{N}_i} w_{ij} \cdot P(l_j = c \vert \mathcal{N}_j)$ (9) with Z a normalizer for the probabilities to sum up to one.

$\mathbf{V} = \{\mathbf{V}^K, \mathbf{V}^U\}$. The inference of the unknown labels based on the known labels is determined by the relational classifier. The order in which unknown class labels are inferred, as well as how the inferred labels influence each other, is coordinated by the CI procedure.

3.1.1. *Relational classifiers*

Table 1 provides a dense summary of the relational classifiers that are applied in the case study in Section 5. Throughout this paper, c will be used to refer to a non-specified class value, and k is an index running over all possible class labels. All relational classifiers included in Table 1 have been implemented using sparse and parallel computation techniques in order to be applicable on massive networks consisting of millions of nodes.[2]

In order to deal with the time dimension that is explicitly present in customer churn, both the CDRN and the NLB relational classifier have been reformulated. When predicting future customer churn in real life, none of the future labels, i.e. at time $t+1$, are known when training a classification model at time t. In other words, the set of vertices with an unknown label equals the set of all vertices, $\mathbf{V} = \mathbf{V}^U$, and the set of vertices with known labels is empty, $\mathbf{V}^K = \emptyset$. Therefore, the unknown labels at time $t+1$ need to be estimated based on the labels at time t, requiring an adjustment to the formulation of the CDRN and the NLB relational classifiers.

The class vector $CV(v_i)$ of a node v_i in the formulation of the CDRN classifier can be defined as the vector of summed linkage weights to the various classes at time t. The reference vector $RV(c)$ of a class c would then be the average of these class vectors for nodes known to be of class c at time $t+1$. However, none of the labels at time $t+1$ are known when training a classifier at time t. Therefore, the reference vectors have to be calculated using data from a previous time frame. For instance, by calculating the average of the class vectors at time $t-1$ for nodes known to be of class c at time t. The class vectors at time t can then be compared to these reference vectors in order to make predictions for time $t+1$.

Similar to the CDRN classifier, the NLB classifier uses the class vectors at time t as independent variables in order to predict the class labels at time $t+1$ by fitting a logistic regression model to the class vectors of nodes known to be of class c at time $t+1$. Since no class labels at time $t+1$ are known on beforehand, the model has to be trained on data from a previous time frame, for instance by using the class vectors at time $t-1$ to predict the class labels at time t.

Finally, the original *SPreading Activation based approach* (SPA) as proposed by Dasgupta et al. [12] and discussed in Section 2.2, is reformulated and split into a relational classifier (SPA RC, see Table 1) and a collective inference procedure (SPA CI, cfr. infra), in order to fit within the framework proposed by Macskassy and Provost [24].

3.1.2. *Collective inference procedures*

Tables 2 and 3 summarize respectively a range of existing and adjusted collective inference procedures that are applied in the experiments in Section 5. The computational complexity of both the original Gibbs Sampling and Iterative Classification formulations appeared to be prohibitive for application on massive social networks consisting of more than a million nodes and links. More specifically, the iterative application of a relational classifier in step 2.(a) of both procedures, running over all the nodes in the network, increases their complexity dramatically. Therefore, an adjusted version of both Gibbs Sampling and Iterative Classification is proposed, which allows them to be applied on very large networks, by making inferences concurrently instead of iteratively. The Gibbs Sampling with Simultaneous Labeling (GSSL) and Iterative Classification with Simultaneous Labeling (ICSL) collective inference procedures schematically work as indicated in Table 3.

Similar to the above described collective inference procedures, the Spreading Activation based Collective Inference procedure (SPA CI) applies a relational learner in each iteration, using the result of iteration i as the input to the next iteration $i+1$. However, whereas Gibbs sampling, relaxation labeling, and iterative classification consist of a specified number of iterations, the SPA CI procedure ends when a stopping criterion is met, consisting of two conditions. The

[2] Nonetheless, the computational complexity of the Network-only Bayes Classifier, as included in NetKit [24] and based on a relational classifier introduced by Chakrabarti et al. [9], appeared to be prohibitive for application on a very large network, specifically in combination with a CI procedure.

Table 2
Summary of original collective inference procedures.

CI	Summary
GS	*Gibbs Sampling* [15] schematically works as follows [24]: [4] 1 Generate a random ordering, O, of vertices in $\mathbf{V}^U$. 2 For elements $v_i \in O$ in order: [(c)] (a) Apply the relational classifier model: $\hat{\mathbf{c}}_i \leftarrow \mathcal{M}_R(v_i)$. (b) Sample a value c_s from $\hat{\mathbf{c}}_i$, such that $P(c_s = c_k \mid \hat{\mathbf{c}}_i) = \hat{\mathbf{c}}_i(k)$. (c) Set $l_i \leftarrow c_s$. 3 Repeat prior step 200 times without keeping any statistics (burnin period). 4 Repeat again for 2000 iterations, counting the number of times each l_i is assigned a particular value $c \in \mathcal{L}$. Normalizing these counts forms the final class probability estimates.
IC	*Iterative Classification* [23] is formulated by Macskassy and Provost [24] as follows: [3] 1 Generate a random ordering, O, of vertices in $\mathbf{V}^U$. 2 For elements $v_i \in O$ in order: [(b)] (a) Apply the relational classifier model, $\hat{\mathbf{c}}_i^{(0)} \leftarrow \mathcal{M}_R$, using all non-null labels (entities which have not yet been classified are ignored). If all neighbor entities are null, then return null. (b) Classify v_i: $l_i = c_k$ and $k = \text{argmax}_j \left(\hat{\mathbf{c}}_i(j)\right)$, where $\hat{\mathbf{c}}_i(j)$ is the *jth* value in vector $\hat{\mathbf{c}}_i$. 3 Repeat for $T = 1000$ iterations, or until no entities receive a new class label. The estimates from the final iteration will be used as the final class probability estimates.
RL	*Relaxation Labeling* [9] is defined by Macskassy and Provost [24] as follows: [2] 1 For elements $v_i \in \mathbf{V}^U$: Estimate l_i by applying the relational model: $\hat{\mathbf{c}}_i^{(t+1)} \leftarrow \mathcal{M}_R(v_i^{(t)})$(10) where $\mathcal{M}_R(v_i^{(t)})$ denotes using the estimates $\hat{\mathbf{c}}_j^{(t)}$ for $v_j \in \mathbf{V}^U$, and t is the iteration count. This has the effect that all the iterations are done pseudo-simultaneously based on the state of the graph after iteration t. 2 Repeat for T iterations. $\hat{c}^T$ will comprise the final class probability estimates.
RL SA	*Relaxation Labeling with Simulated Annealing* applies simulated annealing for the resulting labels to converge, and substitutes Equation 10 by the next expression: $\hat{\mathbf{c}}_i^{(t+1)} = \beta^{(t+1)} \cdot \mathcal{M}_R(v_i^{(t)}) + (1 - \beta^{(t+1)}) \cdot \hat{\mathbf{c}}_i^{(t)}$ (11) with $\beta^0 = k$, and $\beta^{(t+1)} = \beta^{(t)} \cdot \alpha$, k a constant between zero and one, and α a decay constant.

Table 3
Summary of adjusted collective inference procedures.

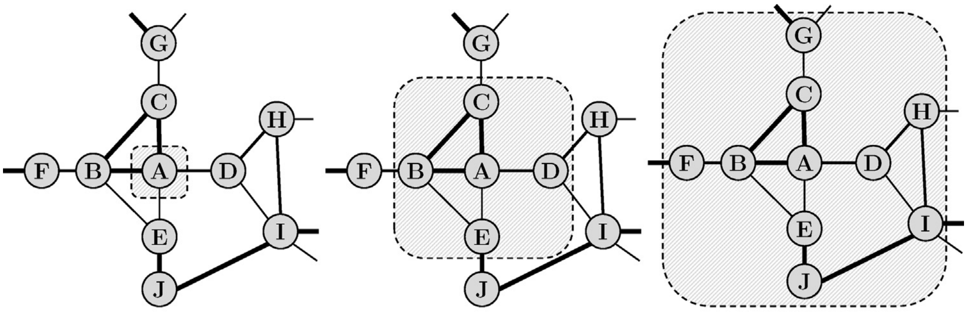
CI	**Summary**
GS SL	*Gibbs Sampling with Simultaneous Labeling* replaces steps 1 and 2 in the above inference scheme by a single step: [1] 1 [(c)] (a) Apply the relational classifier model: $\hat{\mathbf{c}_i^{t+1}} \leftarrow \mathcal{M}_R(v_i^t)$, where $\mathcal{M}_R(v_i^{(t)})$ denotes using the estimates $\hat{\mathbf{c}}_j^{(t)}$ for $v_j \in \mathbf{V}^U$, and t is the iteration count. This has the effect that all the iterations are done pseudo-simultaneously based on the state of the graph after iteration t. (b) Sample a value c_s from $\hat{\mathbf{c}}_i$, such that $P(c_s = c_k \mid \hat{\mathbf{c}}_i) = \hat{\mathbf{c}}_i(k)$. (c) Set $l_i \leftarrow c_s$.
IC SL	*Iterative classification with Simultaneous Labeling* replaces steps 1 and 2 in the original IC scheme by a single step: [1] 1 [(b)] (a) Apply the relational classifier model, $\hat{\mathbf{c}}_i^{(t+1)} \leftarrow \mathcal{M}_R(v_i^{(t)})$, using all non-null labels (entities which have not yet been classified are ignored), and where $\mathcal{M}_R(v_i^{(t)})$ denotes using the estimates $\hat{\mathbf{c}}_j^{(t)}$ for $v_j \in \mathbf{V}^U$, and t is the iteration count. This has the effect that all the iterations are done pseudo-simultaneously based on the state of the graph after iteration t. If all neighbor entities are null, then return null. (b) Classify v_i: $l_i = c_k$ and $k = \text{argmax}_j \left(\hat{\mathbf{c}}_i(j)\right)$, where $\hat{\mathbf{c}}_i(j)$ is the *jth* value in vector $\hat{\mathbf{c}}_i$.
SPA CI	The *Spreading Activation Collective Inference* procedure based on the SPA approach described by Dasgupta et al. [12] can be defined as follows: [3] 1 For $v_i \in \mathbf{V}^U$, initialize the prior: $\hat{\mathbf{c}}_i^{(0)} \leftarrow \mathcal{M}_L(v_i)$, where $\hat{\mathbf{c}}_i$ is defined as above in the Gibbs sampling algorithm. 2 For elements $v_i \in \mathbf{V}^U$: Estimate l_i by applying the relational model: $\hat{\mathbf{c}}_i^{(t+1)} \leftarrow \mathcal{M}_R(v_i^{(t)})$(12) where $\mathcal{M}_R(v_i^{(t)})$ denotes using the estimates $\hat{\mathbf{c}}_j^{(t)}$ for $v_j \in \mathbf{V}^U$, and t is the iteration count. This has the effect that all the iterations are done pseudo-simultaneously based on the state of the graph after iteration t. 3 Repeat while [(c)] (a) $\sum(\hat{\mathbf{c}}_i^{(t+1)} - \hat{\mathbf{c}}_i^{(t)}) > \Delta c_{min}$, with Δc_{min} the minimum overall difference in predicted class labels, (b) OR $\#(\hat{\mathbf{c}}_i^{(t+1)} > 0) > \#(\hat{\mathbf{c}}_i^{(t)} > 0)$, (c) AND $t < T_{max}$, with T_{max} the maximum number of iterations. The resulting $\hat{c}^{t_{end}}$ will comprise the final class probability estimates.

procedure ends when (1) the set of active nodes is not extended, and (2) the amount of energy that is spread, i.e. the overall change in the assigned labels, is smaller than a predefined amount E_t. When combining SPA CI with SPA RC, the stopping criterion will be met since the amount of energy that is passed is reduced by the SPA RC classifier in each iteration, by application of the diffusion coefficient $d \in (0, 1)$. However, convergence is not guaranteed when combining SPA CI with any of the other relational classifiers defined in Section 3. Therefore, similar to the RL procedure, a simulated annealing approach with a predetermined maximum number of iterations is applied when combining SPA CI with any of the other relational classifiers. Eq. (12) is then replaced by Eq. (11).

3.2. Non-relational learning with network variables

An alternative approach to the relational classifiers and collective inference procedures discussed in the previous sections, exists in transforming the information that is contained within the social network structure into a set of *network variables* or attributes. These network variables can then be used by traditional, non-relational data mining techniques, yielding a CCP model of type two or three according to the framework of Fig. 1. Methods that transform a relational representation of a learning problem into a propositional, feature-based or attribute-value representation are known as propositionalization or featurization approaches [21]. In a customer churn prediction setting a range of network variables can be defined, such as the aggregate number of connections between a subscriber and previously churned subscribers, the total time called to churners, etc. These network variables can be derived from the call graph, or directly from the call detail records as shown in Fig. 1.

Typically, an important fraction of the explanatory variables in a CCP model are usage statistics [40], which aggregate information contained within call detail records, such as the number of contacts (i.e. neighbors in the call graph) and the number of contacts that are subscribers of a competing operator. In order to make a clear



(a) Local variable.

(b) First order network variable.

(c) Second order network variable.

Fig. 3. Schematic representation of the scope of a local (left panel) and a first (middle panel) and second (right panel) order network variable related to instance *A*. (a) Local variable, (b) first order network variable and (c) second order network variable.

distinction between a network variable and a non-network or *local* variable, a formal definition is introduced:

Defintion 4. **A network variable** related to instances or objects aggregates information that is contained within a graph or network structure and makes a differentiation in the destination of outgoing links or the origin of incoming links. A network variable of order *i* aggregates information related to an instance or object contained within its order *i* neighborhood.

Defintion 5. **A local variable** represents information related to instances or objects that are treated as isolated entities with unspecified connections to the outside world. A local variable is a network variable of order 0.

Fig. 3 schematically represents the difference between a local variable and a network variable of order 1 and 2, which is in fact the part or range of the network that is *visible* and summarized by the variable. According to Definitions 4 and 5, a variable such as the number of contacts of a customer is a local variable, since it aggregates information from the call graph but does not differentiate between types of contacts. On the other hand, the number of contacts of a customer that are subscribers of a competing operator is a network variable, since a differentiation is made between the origin of incoming and the destination of outgoing links.

The advantage of a propositionalization approach is that it allows to use powerful, accustomed, non-relational modeling techniques. The disadvantage of this approach is that it does not fully take advantage of the possibilities offered by networked data, and that valuable information may be lost in the conversion of the network into attributes.

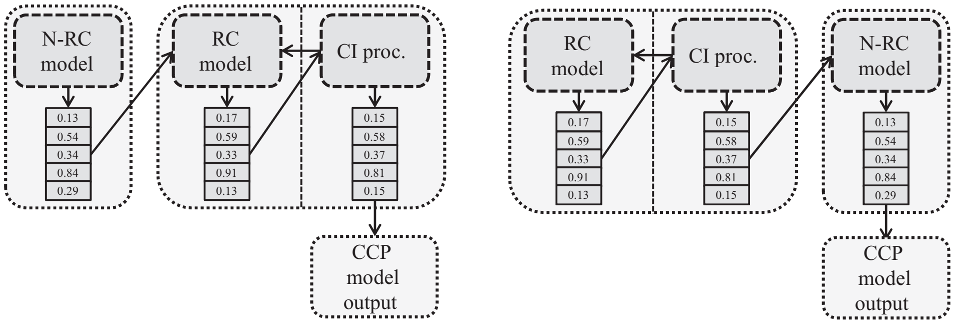
3.3. Combining relational and non-relational classifiers

In this section four approaches are presented to combine a relational and non-relational classifier, constituting the base classifiers of an *ensemble* or combined CCP model of type 4 according to the framework of Fig. 1. For an extensive overview on ensemble learning for classification, one may refer to, e.g. Hastie et al. [18].

A first approach to combine a non-relational classifier with a relational classifier and optionally a collective inference procedure in a unified setup follows from the definitions of the RC and CI algorithms; the probabilities or scores resulting from a non-relational classification model can be used to initialize the labels of the nodes in the network. The non-relational model is used in this setup to provide a first estimate of the class labels of the nodes in the network. Subsequently, the relational classifier and the collective inference procedure are added as a second model layer on top of the non-relational classifier, with the intention to refine and improve the results of the non-relational model by using the information that is incorporated within the networked data.

Conversely, the predicted probabilities by the relational model can be included in the non-relational model as an explanatory variable. In this approach, the non-relational model constitutes a second model layer on top of the relational model. In fact, this approach can be regarded as an *automated* propositionalization of the network, leading to a single network attribute that is not explicitly defined but nonetheless aggregates information that is contained within the network. Both *cascading* approaches or *sequential* model setups are schematically represented in Fig. 4.

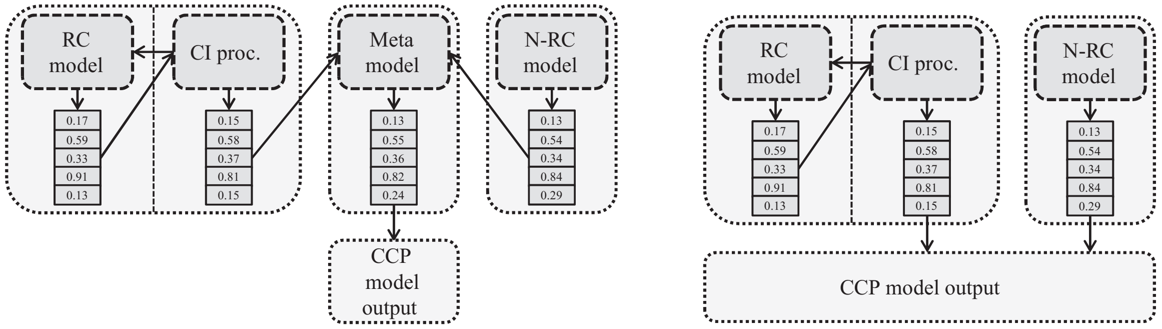
An alternative approach called *stacking* exists in combining the output scores of the relational and non-relational classifier by learning a model on top of these two models. The stacking approach uses the probabilities resulting from the non-relational classification model and the relational classification model, whether or



(a) Non-relational model as input to relational model.

(b) Relational model as input to non-relational model.

Fig. 4. Schematic representation of two approaches to combine a collective inference procedure and a relational and non-relational classification model. (a) Non-relational model as input to relational model and (b) relational model as input to non-relational model.



(a) Stacking approach.

(b) Parallel model setup.

Fig. 5. A stacking approach (left panel) and a parallel, non integrated setup (right panel) to combine a non-relational model, a relational model, and a collective inference procedure. (a) Stacking approach and (b) parallel model setup.

not in combination with a CI procedure, as input variables. This approach is schematically represented in Fig. 5(a). In principle, any non-relational classification technique could be applied to build a second model layer on top of the relational and non-relational classification models.

Finally, the relational and the non-relational model can also be applied in a parallel, non-integrated setup, i.e. by selecting customers indicated to have a high probability to churn either by the non-relational model or by the relational model (or by both models), as shown in Fig. 5(b). This approach is called *voting*, and a customer is classified to be a churner when either the relational or the non-relational CCP model classifies a customer as a churner. Remark that when the base classifiers result in a probability estimate or a continuous output score, with a higher score incorporating a higher probability to churn, the cutoff value for a customer to be classified as a churner can be set independently for the base classifiers. In fact, as indicated in Section 2.3, these cutoff values need to be optimized in order to maximize the resulting lift or profit, involving a combinatorial optimization problem.

4. Modeling non-Markovian network effects

The skewed class distribution of churners vs. non-churners causes relational learning techniques to experience difficulties in learning powerful models. In the case studies presented in the next section, only a relatively small fraction of future churners are directly connected to previous churners. However, a much larger fraction of the churners is connected indirectly. Inspired by the finding of higher order, non-Markovian network effects on the happiness and health of individuals [10], in this section we introduce an approach to incorporate higher network effects in relational learners without the need to adjust the inner workings of these techniques.

4.1. Non-Markovian network effects

The order n network neighborhood of a particular node was defined in Section 2. The concept of network neighborhood can be applied straightforward to a set of nodes as well, as the union of all the neighborhoods of the nodes in the set. In this section the *exclusive* order n network neighborhood of a particular node, which is applicable to a set of nodes as well, is defined as follows:

Defintion 6. The exclusive order o network neighborhood $\mathcal{N}_i^{o,e}$ of node v_i is defined as the subset of nodes of the set of all nodes in the network **V** that are connected to node v_i with the number of links of the shortest path connecting the nodes in the neighborhood and the node v_i exactly equal to the order o.

Nodes pertain to the exclusive order o network neighborhood with the lowest order o when applying this definition to a set of nodes. For instance, when node A is a first order neighbor of node B and a second order neighbor of node C, then A only pertains to the exclusive first order network neighborhood of the set of nodes $\{B, C\}$, and not to the exclusive second order neighborhood. Fig. 6 provides an indication of the existence of higher order network effects in a customer churn setting, resulting from the analysis of a call graph which will be described into detail in Section 5. The figure plots the effect on the prior probability to churn in the exclusive order x neighborhood of the churners in the network, as a function of the order x. The grey bars indicate the fraction of the customer base **V** that is included in the exclusive order x network neighborhood $\mathcal{N}^{x}_{\mathbf{V}^{t-1}_c}$ of the set of customers $\mathbf{V}^{t-1}_c$ that churn in time frame $t-1$. The black line indicates the churn rate in time frame t in this neighborhood, and the dashed black line represents the base churn fraction in the entire customer base in time frame t. As can be observed from this plot, customers that are direct, first order neighbors of churners clearly have a much larger probability to churn than random customers. However, also customers that are second and even higher order neighbors of churners display an increased prior probability to churn. As the order of the neighborhood increases, the network effect on the probability to churn decreases, and as of order five the effect has entirely disappeared. Fig. 6 provides an important indication of the existence of higher order effects on the churn behavior of customers. The order 1 effect in Fig. 6 is exactly what has been called homophily, assortativity, or relational autocorrelation in Section 3.

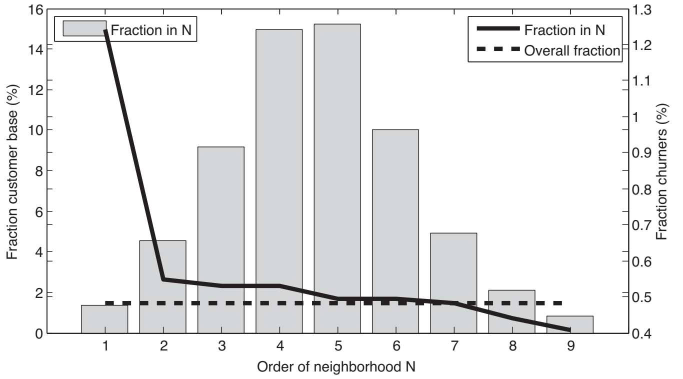


Fig. 6. Effect on the prior probability to churn of the order x. The grey bars indicate the fraction of the customer base **V** that is included in the exclusive order x network neighborhood $N^{x}_{\mathbf{V}^{t-1}_c}$ of the set of customers that churn in time frame $t-1$, $\mathbf{V}^{t-1}_c$. The black line indicates the churn rate in time frame t in this neighborhood, and the dashed black line represents the base churn fraction in the entire customer base.

Notice that the set of customers in the order x neighborhood does not contain the customers in the order $x-1$ neighborhood, since the concept of the exclusive order x neighborhood is applied. Furthermore, in order to cancel out sequences of first order effects instead of *pure* higher order effects,[3] the higher order neighbors of churners in lower order neighborhoods are not taken into account. This is important in order to isolate and clearly distinguish between the effects of the neighborhood order. However, whether higher order effects are in fact sequences of first order effects or not, it is relevant to take these effects into account. Fig. 6 shows that for an increasing order of the exclusive neighborhood the amount of subscribers in the neighborhood increases as well, until order five. Although a smaller fraction of the subscribers in the exclusive neighborhood of order two churn, the absolute number of churners in this neighborhood is much higher than the absolute number of churners in the order one neighborhood, which reconfirms the importance of taking into account higher order network effects.

4.2. The weight product to calculate higher order network connections

When applying relational classifiers that restrict the impact of the network to the first order neighborhood in combination with collective inference procedures, the impact of a particular node partially propagates or spreads throughout the network and reaches nodes in higher order neighborhoods. This is the result of the iterative application of first order neighborhood relational classifiers, since subsequent first order effects constitute a higher order effect. However, as will be shown in the experimental results section, the application of collective inference procedures may deteriorate the performance of the resulting classification model. Nonetheless, as indicated in the above section, the impact of higher order neighborhood nodes appears relevant to be incorporated within a CCP model. Therefore, this section introduces a novel approach to *upgrade* the order of the weight matrix in order to include higher order nodes within the local neighborhood that is taken into account by the relational learners. Higher order nodes are transformed into first order neighborhood nodes, with appropriate values assigned to the respective weights. This allows relational classifiers to model non-Markovian network effects without the need to apply a collective inference procedure and without the need to adjust the inner workings of the relational learning techniques.

The problem of upgrading the order of a weights matrix is closely related to the problem of computing the *minimum-plus product* or *distance product* of a distance matrix in the *All Pairs Shortest Path* (APSP) problem [41].

Defintion 7. The **standard matrix product** or the plus-times matrix product $C=A\cdot B=(c_{ij})$ for $n\times n$ matrices $A=(a_{ij})$ and $B=(b_{ij})$ is defined by:

$$c_{ij}=\sum_{k=1}^{n}a_{ik}\cdot b_{kj}. \tag{13}$$

Defintion 8. The **distance matrix product** or the minimum-plus matrix product $C=A\star B$ is defined similar to the plus-times product $C=A\cdot B$, but with the summation operator replaced by the minimum operator and the product operator replaced by the summation operator:

$$\begin{aligned} c_{ij} &= a_{ij}\star b_{ij}\\ &= \min_k(a_{ik}+b_{kj}). \end{aligned} \tag{14}$$

The distance product $D^{\star 2}=D\star D^T$, with D a distance matrix and elements d_{ij} representing the first order distance between nodes v_i and v_j, results in a matrix $D^{\star 2}$ with elements $d_{ij}^{\star 2}$ equal to the shortest distance between nodes spanning *exactly* two edges. The matrix $min(D, D^{\star 2})$ yields the shortest distance between nodes spanning *maximum* two edges. In order to upgrade the order of the weights matrix, we define a matrix operation that is equivalent to the distance product but applies to weight matrices, by assuming that weights are equivalent to the inverse of distance, i.e. the higher the value of a weight w_{ij}, the closer two nodes are related:

Defintion 9. The **weight matrix product** or the maximum-of-times-divided-by-plus matrix product $C=A\circledast B$, is defined as follows:

$$\begin{aligned} c_{ij} &= a_{ij}\circledast b_{ij}\\ &= \max_k\left(\frac{a_{ik}\cdot b_{kj}}{a_{ik}+b_{kj}}\right). \end{aligned} \tag{15}$$

Theorem 1. *The weight matrix product $W^{\circledast 2}=W\circledast W^T$ defined by Definition 9 is the equivalent matrix operation for weight matrices of the distance matrix product for distance matrices $D^{\star 2}=D\star D^T$ defined by Definition 8, assuming weights representing a link in a graph are equivalent to the inverse of distances associated with links in a graph:*

$$(w_{ij})\circledast(w_{ij})^T=(w_{ij})^{-1}\star(w_{ij})^{-1,T} \tag{16}$$

or,

$$(d_{ij})\star(d_{ij})^T=(d_{ij})^{-1}\circledast(d_{ij})^{-1,T} \tag{17}$$

A formal proof of this theorem can be found in appendix. The maximum of the weight matrix W and the weight product of the weight matrix $W^{\circledast 2}$, i.e. $max(W, W^{\circledast 2})$, yields an upgraded weight matrix that incorporates second order nodes *as if* first order nodes with appropriate weights assigned to the links representing second order connections. p subsequent applications of the weight product and selection of the maximum weight for each relation between two nodes in each step, yields the weight matrix of order p.

The equivalence between the weight and distance product is illustrated in Fig. 7, which plots a weighted version of the simple example network used in the previous section with first and second order links to node A, and the equivalent distance networks.

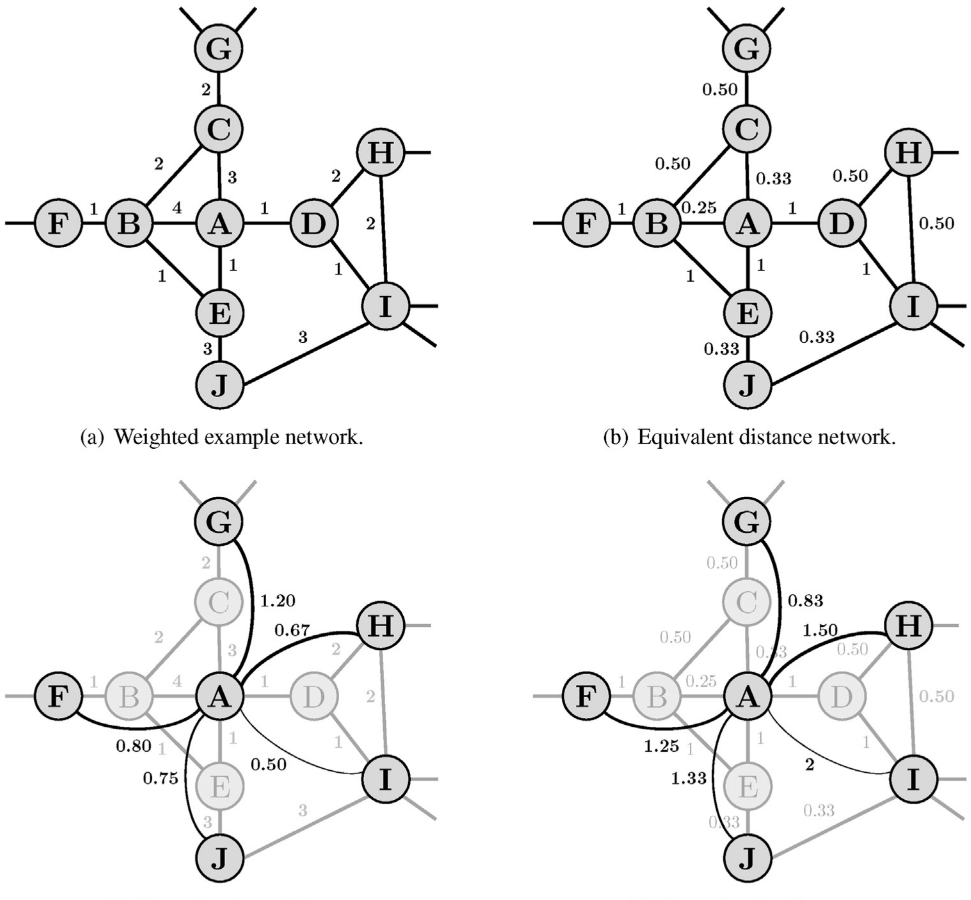
5. Case studies

This section presents the application of the presented techniques in two real life case studies, concerning a prepaid and postpaid customer segment.

5.1. Data set and experimental setup

CDRs of voice to voice calls for both a prepaid and a postpaid customer segment were provided by an anonymous European telco operator. The time range of the CDR data covers a period of five months, which will be denoted M1 to M5. Churn labels, indicating the exact date when customers churned, were available for this period and one month prior and after. Based on previous studies and extensive discussions with telco experts, CDRs are converted into a network by defining nodes as subscribers of the operator, and edges as the total number of seconds of voice to voice calls between these subscribers. The resulting network is represented by

[3] For instance, assume that a churning subscriber in time frame $t-1$ causes a connected subscriber to churn in time frame t. Within time frame t, this churned neighbor causes one of his connections to churn, resulting in a *second* order effect, which is in fact a sequence of *two first order effects*. This also illustrates the impact of the time frame. Higher order effects resulting from sequential first order effects can be expected to have a larger impact for longer time frames t.



(a) Weighted example network.

(b) Equivalent distance network.

(c) Second order weights.

(d) Second order distances.

Fig. 7. The weighted example network around node *A* and the equivalent distance network with first (upper panels) and second order (lower panels) weights (left panels) and distances (right panels). Remark that the width and length of links in these networks are not representative for the actual values of the edge weights or distances. (a) Weighted example network, (b) equivalent distance network, (c) second distance network and (d) second order distances.

an undirected weight matrix, which was preferred over a directed matrix to reduce computational complexity of the relational learning process. The number of subscribers in the customer base is 1, 673, 724 for the prepaid segment and 1, 226, 286 for the postpaid segment. The number of edges connecting customers in the social network derived from the CDR data equals 2, 414, 945 for the prepaid segment and 3, 706, 384 for the postpaid segment. The class distribution is very skewed, which is typically the case for customer churn data sets as discussed in Section 2, with on average only 0.52% and 0.57% of the customers in respectively the prepaid and postpaid segment that churn each month. Furthermore, 119 and 58 attributes, both local and network variables, were provided for each customer in respectively the prepaid and postpaid customer base.

Three months of data, M2 to M4, have been selected to induce a call graph and training labels, indicating which subscribers churned during this period. The call graph and the training labels are used by the relational classifiers to predict the test labels, i.e. to predict which subscribers will churn in month M5 immediately after the period M2 to M4. As indicated in Section 3, in order to correctly handle the time dimension, both the CDRN and NLB classifier require a previous time frame in order to calculate the class vectors. This previous time frame consists of the months M1 to M3, with test labels of month M4. Furthermore, the data attributes of month M2 and churn labels of month M3 were used to train a non-relational model. The induced model was then applied to predict churn in month M5 using the data attributes of month M4. Remark that the predictions that are made for month M5 result in a strict out-of-time evaluation of the performance of the induced models, which provides a correct indication of how a model would perform in a real life setting, since no data of month M5 have been used in training the models.

5.2. Results and discussion

A selection of non-relational classification techniques is evaluated with and without network variables to assess the impact of network variables on the predictive power of a CCP model. Next, the applicability of relational learning techniques for CCP modeling is assessed, including the impact of collective inference procedures and upgrading the network order, and compared to non-relational classifiers. Finally, ensembles of relational and non-relational classifiers are experimentally evaluated and compared to the stand-alone base classifiers.

5.2.1. Non-relational classification with network variables

Non-relational CCP models were built by applying five non-relational classification techniques on the provided data attributes, i.e. Alternating Decision Trees (ADT) [14], Bagging (Bag) [5], Random Forests (RF) [6], Bayesian Networks (BN), and Logistic Regression (Logit). The first four of these techniques constitute the top ranked non-relational classifiers for churn prediction in the telco industry as found in a large benchmarking experiment reported in Verbeke et al. [39], by evaluating the experiments using either the maximum profit measure or top decile lift. Logistic regression on the other hand is considered to be a general industry

Table 4
Results of the non-relational classification techniques with and without network variables (NV) in terms of lift.

	Prepaid					Postpaid				
	Logit	ADT	RF	BAG	BN	Logit	ADT	RF	BAG	BN
Lift at 0.5%										
Without NV	4.35	**4.78**	3.43	**5.74**	**4.81**	4.30	3.34	1.99	2.27	6.27
With NV	**4.94**	4.75	**3.47**	4.85	4.37	**5.84**	**5.56**	**2.88**	**4.52**	**6.83**
Lift at 1%										
Without NV	3.93	3.64	2.70	**4.28**	**4.16**	3.56	3.39	1.98	3.33	5.40
With NV	**3.99**	**3.69**	**2.74**	3.99	3.82	**5.32**	**5.17**	**2.50**	**3.58**	**5.84**
Lift at 5%										
Without NV	3.11	**2.82**	1.64	2.71	3.13	2.44	3.43	1.79	3.22	3.67
With NV	**3.14**	2.08	**1.67**	**3.09**	**3.19**	**3.82**	**3.71**	**1.82**	**3.69**	**3.75**
Lift at 10%										
Without NV	2.72	**2.51**	1.30	2.25	2.80	1.86	2.74	**1.63**	2.71	2.93
With NV	**2.74**	1.93	**1.32**	**2.78**	**2.82**	**3.06**	**3.41**	1.55	**3.19**	**2.97**

standard, and was not found to be statistically significantly outperformed in the benchmarking experiments.

The data attributes provided by the telco operator contain both local and network variables as defined in Section 3.2. Examples of network variables are the number of contacts between a customer and previous churners, the number of contacts between a customer and customers of particular competing operators, etc. Non-relational CCP models were induced using the five selected classifiers on the top 25 ranked attributes, excluding network variables in the first series of experiments and including network variables in the second series. Identical to a procedure applied in Verbeke et al. [39], the attributes were ranked using a chi squared filter to retain the most relevant attributes and facilitate learning. Comparing the results of the first and the second series of experiments allows to assess the impact of network attributes on the predictive power of a non-relational CCP model.

Table 4 reports the results of the experiments in terms of top 0.5%, 1%, 5%, and 10% lift for the prepaid and postpaid case study. As can be seen from the table, including network variables clearly boosts the performance of the non-relational classification model in the postpaid case study. The models including network variables consistently outperformed the models without network variables. The results of the prepaid case study on the other hand are less conclusive about the impact of network variables, but indicate a slight improvement in lift for larger top fractions (top 5% and 10%).

In the postpaid case study three different techniques yield the highest lift depending on the selected top fraction. This illustrates the shortcomings of using lift as a measure to evaluate classification models in a customer churn prediction setting, and motivates the application of the MPC measure as introduced in Section 2.3. When applying the MPC measure to evaluate the predictions of the non-relational classification models, network variables are found to boost the profit per customer, since fractions roughly between 3% and 10% are selected, for which higher lift is obtained when including network variables, as indicated in Table 4.

Furthermore, the results in Table 4 show that the models in the postpaid case study generally obtain better predictive power than the models in the prepaid case study. In a prepaid setting no subscription records are available and thus less information is available to predict churn, yielding lower lift figures. Moreover, in the postpaid case study a significant fraction of churn is explained by an attribute indicating the end-of-contract, which is not available in the prepaid case. The Logit model with network variables will be used in the next sections as the base non-relational model, because of its widespread use in the industry as well as in the research community as a benchmark model.

5.2.2. *Relational learning for customer churn prediction*

Table 5 reports the results of the relational classifiers presented in Section 3 for both the prepaid and the postpaid case study. Except for lift at small top fractions in the prepaid case, the relational classifiers generally yield weaker predictive power than the non-relational classifiers. This is not surprising given the fact that a much smaller amount of information is used to build these models. Only the amount of communication between the customers of the operator and a label indicating churn serves as input to the relational learners. Incorporating information in the relational learning process related to communication with customers of competing operators, as included in the form of network variables in the non-relational models discussed in the previous section, may be an interesting topic for future research to improve the performance of stand-alone relational models.

However, although less powerful, relational learners may be useful since they detect different segments or types of churners than non-relational classifiers. Fig. 8 plots the fraction of the churners detected by a non-relational model (Logit with network variables) that is not detected by the relational models (left panel), and vice versa (right panel), as a function of the selected top fraction of customers. The selected top fraction of customers on the *x*-axis is the same for both the non-relational and the relational models. For instance, when selecting the top 10% of customers with the highest predicted probabilities to churn as indicated by the relational models, then on average about 80% of these churners are not included in the top 10% selected by the non-relational model.

The reason why a relational and a non-relational model detect different segments or types of churners, may be explained by the fact that only a limited fraction of churn events is accounted for by social network effects. These *social churners* are the only churners that can possibly be detected by the relational learners, which also explains the weaker lift of these models for large top fractions (top 5% and 10%) as reported in Table 5. In case of the postpaid segment, the fraction of social churners is even smaller than in case of the prepaid segment, explaining the difference in lift figures between both segments. On the other hand, because the social churners only constitute a relatively small fraction within the already small fraction of churners, non-relational classifiers are not able to detect this type of churn, since the related patterns are not sufficiently or not explicitly present in the attribute value representation of the data in order for a non-relational learner to incorporate such a pattern in the resulting classification model. Non-relational learners appear to incorporate other patterns in the model, which are more prevalent in the data and have better predictive power.

The complementarity of relational and non-relational classifiers with regards to their ability to detect different segments of

Table 5
Results of the experiments for the two case studies (prepaid and postpaid), combining a relational classifier (RC) and a collective inference (CI) procedure for the network neighborhood order (NO) equal to one and two. The highest lift per segment is indicated in bold, and the overall highest lift per top fraction is underlined.

	CI	Prepaid						Postpaid					
		–	GS SL	RL	RL SA	IC SL	SPA CI	–	GS SL	RL	RL SA	IC SL	SPA CI
	RC	**Lift at 0.5%**											
NO = 1	WVRN	4.58	2.40	3.48	3.54	1.51	2.74	2.79	1.31	1.81	2.00	1.43	1.50
	CDRN	4.58	1.06	1.00	1.00	1.00	4.16	2.79	1.21	1.07	1.06	1.00	2.65
	NLB	4.58	1.00	1.00	1.00	1.00	1.00	2.79	1.00	1.01	1.01	1.00	1.02
	SPA RC	**4.91**	1.00	1.73	1.56	1.00	1.50	**3.69**	1.00	1.19	1.17	1.00	1.09
NO = 2	WVRN	**4.65**	2.29	3.16	3.34	1.31	2.80	**3.15**	1.25	1.37	1.46	1.15	1.29
	CDRN	**4.65**	1.03	1.37	1.37	1.00	3.41	**3.15**	1.12	1.06	1.06	1.00	2.22
	NLB	**4.65**	1.33	1.37	1.37	1.00	2.65	**3.15**	1.00	1.00	1.00	1.00	1.00
	SPA RC	4.02	1.04	1.53	1.57	1.00	1.47	3.12	1.00	1.23	1.20	1.00	1.08
Lift at 1%													
NO = 1	WVRN	**4.24**	1.95	2.62	2.70	1.25	2.18	2.76	1.23	1.59	1.68	1.21	1.33
	CDRN	3.70	1.05	1.00	1.00	1.00	3.41	2.76	1.11	1.07	1.06	1.00	2.39
	NLB	**4.24**	1.00	1.00	1.00	1.00	1.00	2.76	1.00	1.01	1.01	1.00	1.02
	SPA RC	3.78	1.00	1.73	1.54	1.00	1.50	**3.31**	1.00	1.17	1.15	1.00	1.09
NO = 2	WVRN	**3.91**	1.89	2.52	2.62	1.16	2.12	**2.89**	1.24	1.31	1.38	1.08	1.25
	CDRN	3.87	1.01	1.37	1.37	1.00	2.74	**2.89**	1.06	1.06	1.06	1.00	1.96
	NLB	**3.91**	1.33	1.37	1.37	1.00	2.46	**2.89**	1.00	1.00	1.00	1.00	1.00
	SPA RC	3.24	1.02	1.53	1.55	1.00	1.47	2.72	1.00	1.19	1.18	1.00	1.08
Lift at 5%													
NO = 1	WVRN	**1.94**	1.56	1.79	1.79	1.05	1.70	**1.96**	1.12	1.23	1.25	1.04	1.18
	CDRN	1.52	1.02	1.00	1.00	1.00	1.77	1.60	1.02	1.07	1.06	1.00	1.61
	NLB	1.71	1.00	1.00	1.00	1.00	1.00	1.57	1.00	1.01	1.01	1.00	1.02
	SPA RC	1.53	1.00	1.58	1.53	1.00	1.50	1.81	1.00	1.16	1.13	1.00	1.09
NO = 2	WVRN	2.44	1.50	1.70	1.74	1.03	1.58	1.78	1.15	1.21	1.22	1.01	1.18
	CDRN	1.56	1.01	1.37	1.37	1.00	1.70	1.62	1.01	1.06	1.06	1.00	1.54
	NLB	**2.48**	1.09	1.37	1.37	1.00	2.08	1.46	1.00	1.00	1.00	1.00	1.00
	SPA RC	2.01	1.00	1.53	1.54	1.00	1.47	**1.81**	1.00	1.16	1.16	1.00	1.08
Lift at 10%													
NO = 1	WVRN	1.45	1.41	1.48	1.46	1.02	1.53	**1.45**	1.09	1.16	1.19	1.02	1.12
	CDRN	1.25	1.02	1.00	1.00	1.00	1.41	1.28	1.01	1.07	1.06	1.00	1.30
	NLB	1.34	1.00	1.00	1.00	1.00	1.00	1.27	1.00	1.01	1.01	1.00	1.02
	SPA RC	1.25	1.00	**1.56**	1.52	1.00	1.50	1.38	1.00	1.15	1.13	1.00	1.09
NO = 2	WVRN	1.68	1.25	1.60	1.63	1.01	1.51	1.55	1.13	1.17	1.18	1.01	1.15
	CDRN	1.27	1.00	1.37	1.37	1.00	1.37	1.31	1.01	1.06	1.06	1.00	1.30
	NLB	**1.92**	1.04	1.37	1.37	1.00	1.81	1.24	1.00	1.00	1.00	1.00	1.00
	SPA RC	1.48	1.00	1.53	1.54	1.00	1.47	**1.57**	1.00	1.16	1.15	1.00	1.08

churners, opens opportunities for combining a relational and a non-relational model to obtain a classification model with increased predictive power. The next section presents the results of the different approaches to combine a relational and a non-relational model as discussed in Section 3.3.

A second main finding of Table 5 concerns the impact of upgrading the network neighborhood order on the predictive power of the relational classification model. As can be seen from Table 5, for small top fractions (0.5 and 1%) the best predictive power was obtained for network neighborhood order equal to one, both in the

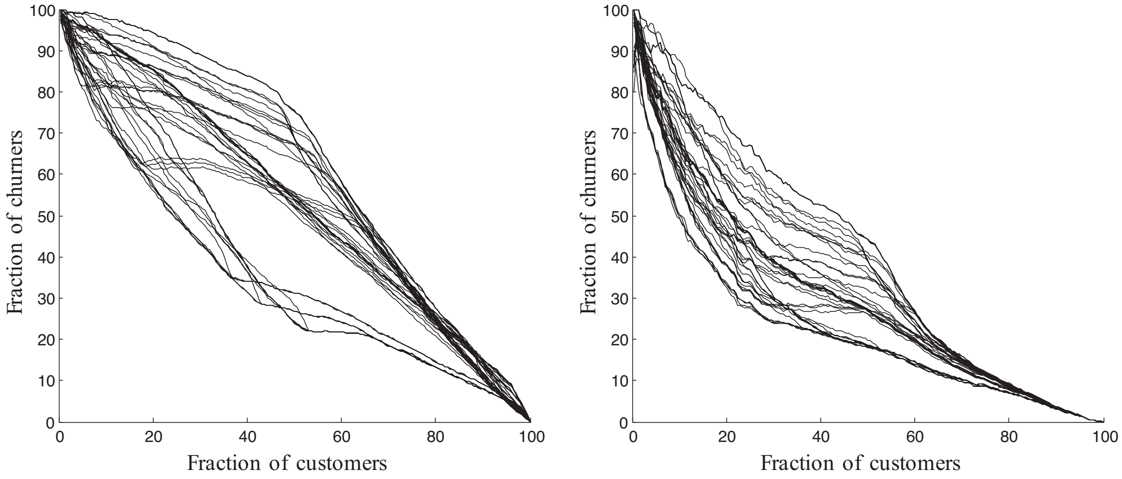
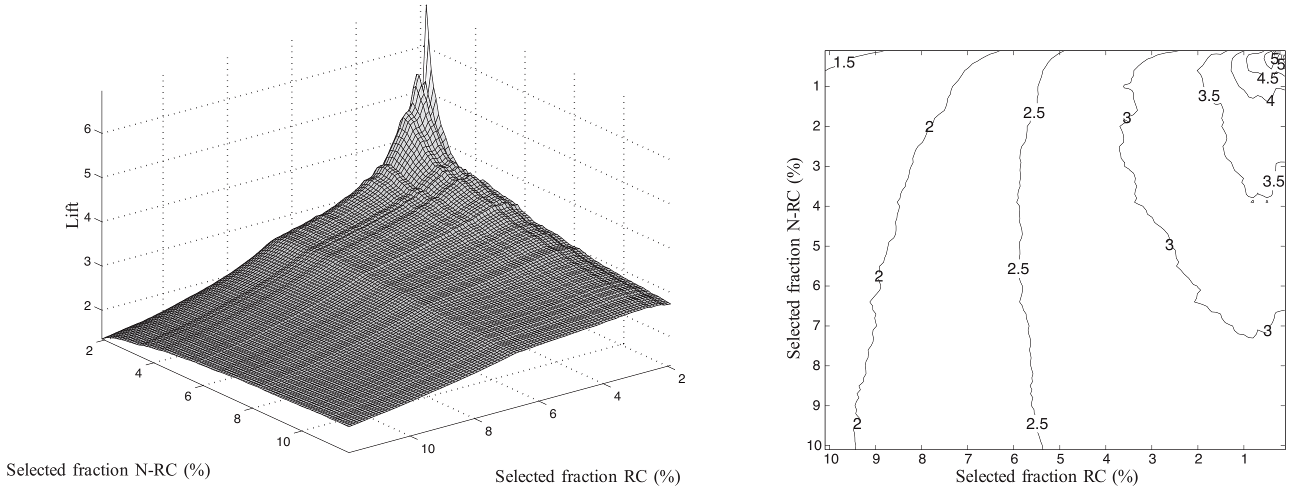


Fig. 8. The fraction of the churners detected by the non-relational model (Logit) that is not detected by the relational classification models reported in Table 5 as a function of the selected fraction of customers with the highest predicted probability to churn (left panel), and vice versa (right panel), for the prepaid segment.

Table 6
Results of the experiments in terms of top 0.5, 1, 5, and 10% lift, for combining a non-relational logistic regression classification model with relational classifiers for a network neighborhood order (NO) equal to one and two, using a parallel model setup. The highest lift per segment is indicated in bold, and underlined if better than the lift of the stand-alone relational and non-relational model.

	Lift at	Prepaid				Postpaid			
		0.5%	1%	5%	10%	0.5%	1%	5%	10%
NO = 1	WVRN	5.08	**4.74**	**3.47**	**2.90**	**5.84**	**5.48**	3.84	3.08
	CDRN	5.08	**4.74**	3.37	2.84	**5.84**	**5.48**	3.84	3.08
	NLB	5.08	**4.74**	3.37	2.89	**5.84**	**5.48**	3.84	3.08
	SPA RC	**5.49**	4.64	3.28	2.85	**5.84**	**5.48**	**3.92**	**3.11**
NO = 2	**WVRN**	5.46	4.67	3.35	2.86	**5.84**	**5.48**	3.88	3.09
	CDRN	5.46	4.67	3.35	2.85	**5.84**	**5.48**	3.88	3.09
	NLB	5.46	4.67	3.35	2.86	**5.84**	**5.48**	3.88	3.09
	SPA RC	5.27	4.31	3.23	2.79	**5.84**	**5.48**	3.87	3.08



(a) Lift curve of parallel model.

(b) Contour plot of lift curve.

Fig. 9. Lift curve (left panel) with contour plot (left panel) of the parallel model as a function of the selected fraction of subscribers of a relational model (WVRN) and a non-relational model with network variables (Logit). (a) Lift curve of parallel model and (b) contour plot of lift curve.

prepaid and postpaid case study. However, for top 5% and 10% in prepaid and top 10% in postpaid, predictive power increased when upgrading the network neighborhood order to two[4].

Upgrading the network neighborhood order induces noise with regards to first order social network effects. This results in decreased predictive power when assessing lift at small top fractions. However, as explained above, only a very small fraction of the churners are social churners and are directly (i.e. within their neighborhood of order 1) connected to previous churners, leading to poor lift at top 5% and 10%. The weight product to upgrade the network neighborhood order, as developed in the previous section, was specifically designed to handle the skewed class distribution and to improve the overall classification performance, i.e. at large top fractions. The increased lift figures for top 5% and 10% as reported in Table 5 indicate that the proposed approach effectively functions, and allows to induce classification models that lead to a better overall classification of the customers at the cost of a small decrease in classification performance with regards to the customers with the highest predicted probabilities to churn.

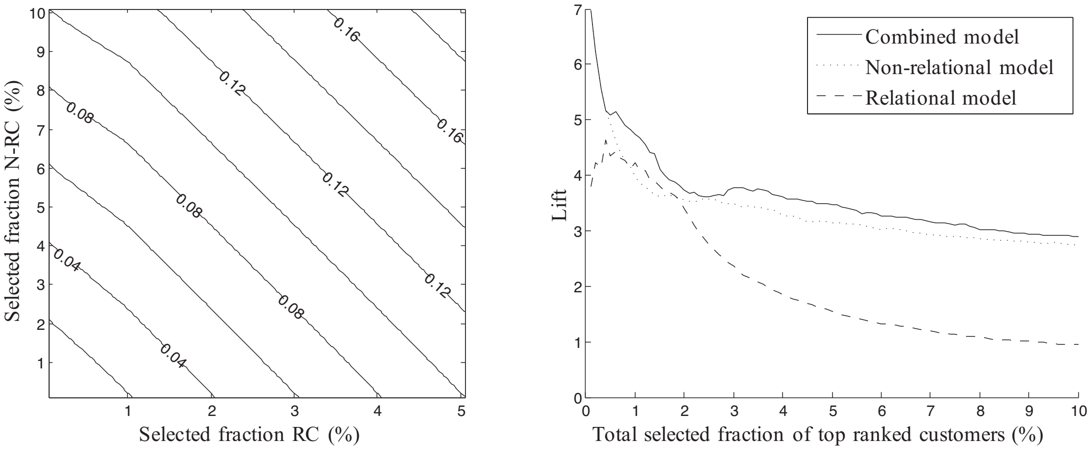
In the postpaid segment only a minor increase in lift is obtained by upgrading the network neighborhood order. On the other hand, including network variables in a non-relational network model consistently improved the predictive power, as shown by the results reported in Table 4 in the previous section. This indicates that a significant fraction of churn is explained by social network effects, but mainly competing operator traffic. This confirms that incorporating such information within a relational model, as indicated above, is a prime issue for future research.

A final finding from Table 5 concerns collective inference procedures, which clearly have a negative impact on the classification performance in a customer churn prediction setting. Therefore in the next section we will not consider CI procedures when combining relational and non-relational classifiers. The bad performance of collective inference procedures is due to the large amount of noise they introduce to the resulting predictions by spreading the impact of each node and smoothing the predictions over a wide part of the network. This in fact stems from their initial intent and use, since these procedures were initially designed for image restoration [9].

5.2.3. Combined relational and non-relational classification model

Both the cascading and stacking approaches as shown in Figs. 4(a), (b), and 5(a) have been found unable to improve the predictive power of the stand-alone non-relational model. The main reason lies in the fact that a local and a network model detect different types of churners. The initialization, automated propositionalization, and stacking approaches favor customers with high *average* probabilities to churn (averaged over the relational and non-relational model). However, churners detected by the network model are assigned a fairly low probability to churn by the

[4] Further upgrading the network neighborhood to order three and higher did not yield improved classification performance.

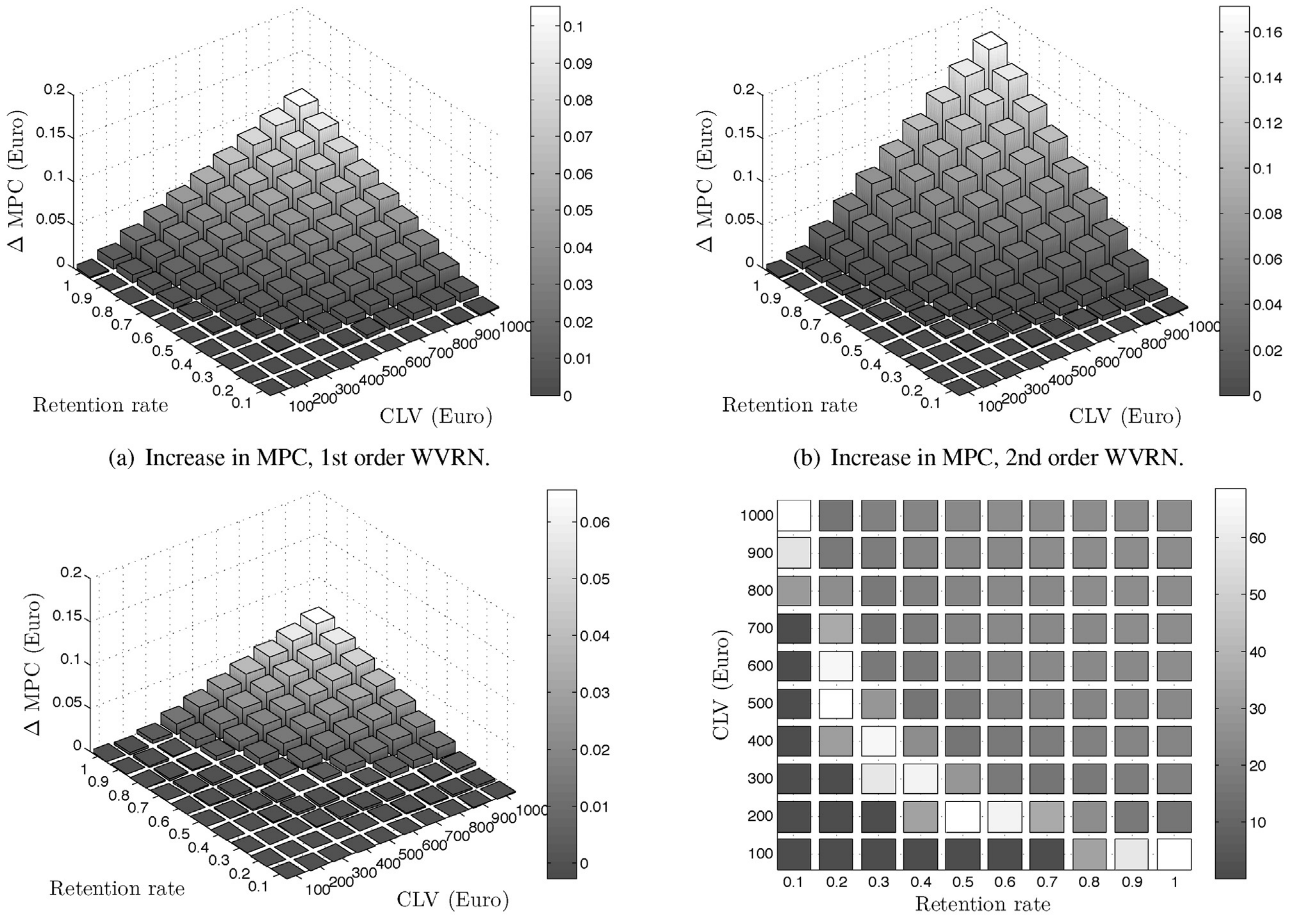


(a) Total selected fraction.

(b) Lift curves.

Fig. 10. Total selected fraction of customers by the parallel model as a function of the selected fractions of the relational and non-relational model (left panel); the lift curves of a non-relational (Logit), a relational (WVRN), and a combined model (Logit-WVRN) for the prepaid case study (right panel). (a) Total selection fraction and (b) lift curves.

non-relational model, and vice versa. This results in a medium average probability to churn, and consequently relatively poor classification power of the integrated combination approaches. The parallel setup however selects customers indicated to have a high probability to churn either by the non-relational model or by the relational model, yielding improved predictive power compared to the stand-alone models. Table 6 reports the results of the parallel combination approach for different relational learners and the Logit non-relational model with network variables. The improvement in predictive power of the combined model in the postpaid case study



(a) Increase in MPC, 1st order WVRN.

(b) Increase in MPC, 2nd order WVRN.

(c) Difference in MPC increase 2nd-1st order.

(d) Percentage increase in MPC.

Fig. 11. Comparison in terms of the maximum profit measure of a parallel model incorporating a traditional non-relational Logit model and a relational WVRN model, with 1st order network effects (top left panel) and with 1st and 2nd order effects (top right panel), compared to the stand-alone Logit model. The bottom right bar plot shows the percentage increase in MPC of the parallel model compared to the stand-alone Logit model. (a) Increase in MPC, 1st order WVRN, (b) increase in MPC, 2nd order WVRN, (c) difference in MPC increase 2nd-1st order and (d) percentage increase in MPC.

is only faint, because the stand-alone relational learner is weaker compared to the prepaid case.

The parallel model approach yields a three dimensional lift curve, indicating the lift for each possible combination of selected top fractions of the relational and non-relational model. The three dimensional lift curve of the combined Logit-WVRN model is shown in Fig. 9(a), and the related iso-lift curves are shown in Fig. 9(b). A two dimensional lift curve, allowing a straight comparison to the lift curves of the constituting stand-alone models, can be derived from the three dimensional curve by selecting the maximum lift for each resulting total fraction selected by the parallel model, i.e. by solving the combinatorial optimization problem mentioned in the final paragraph of Section 3.3. The total selected fraction differs from the sum of the selected fractions of the base models because some overlap exists between these two fractions, as shown in Fig. 10(a). Fig. 10(b) plots the lift curves of the non-relational Logit model, the WVRN relational model, and the combined parallel model Logit-WVRN for the prepaid case study. From this figure can be seen that the combined model clearly improves the predictive power of the stand-alone relational model and the stand-alone non-relational model.

Although the gain in lift appears to be minor, it may well result in a significant increase in profit. Fig. 11 plots the increase in MPC of the combined Logit-WVRN model for NO = 1 (top left panel) and for NO = 2 (top right panel), compared to the stand-alone Logit model with network variables for the prepaid case. Depending on the values of the retention rate and the CLV, and with $\delta = 1$ and $c = 0.05$, the increase in MPC ranges between 0 and 18 cents per customer and per retention campaign, which may yield significant profit gains for customer bases typically consisting of millions of customers, and with retention campaigns typically executed each month of the year.

Furthermore, Fig. 11(c) shows the difference in maximum profit between the parallel model with and without second order effects. It is found that including non-Markovian social network effects within relational classification for customer churn prediction generates additional profits because of improved classification power, which, depending on the value of the retention rate and the CLV, amount to 6 cents per customer in the customer base, per retention campaign. Finally, Fig. 11(d) demonstrates to what extent these novel relational learning techniques improve the traditionally used CCP models, showing the percentage increase in MPC of the parallel model compared to the stand-alone Logit model. The profit gains are in the order of 20–30 percent, and even higher for small absolute values of the MPC, or equal to zero when the MPC equals zero (i.e. when a retention campaign would bear loss instead of profit).

The optimal fraction of included customers in the retention campaign when maximizing the profit, ranges between 0 and 1% for very small values of the retention rate and CLV, and up to 15% for high retention rates and CLV. Hence, more customers should be included in a retention campaign when customers are retained more easily (i.e. when the retention rate is higher), and when it is more profitable to retain customers (i.e. when the average CLV is higher). For the same reason the absolute difference in MPC between the parallel model and the stand-alone non-relational Logit model increases for a higher retention rate and CLV.

6. Managerial insights and conclusions

This study develops a range of new and adapted relational learning algorithms for customer churn prediction using social network effects, designed to handle the massive size of the call graph, the time dimension, and the skewed class distribution typically present in a customer churn prediction setting. Furthermore, an innovative approach to incorporate non-Markovian network effects within relational classifiers is presented, i.e. the weight product, which allows to upgrade the network neighborhood order of the weight matrix. The weight product can be applied complementary to existing relational classifiers, and is shown to be the equivalent operation for weighted networks to the distance product for distance networks. Finally, a novel parallel model setup to combine a relational and non-relational classification model is introduced, which selects a top-fraction of the customers with the highest predicted probabilities to churn of both models. All the techniques that are developed in this study may have a broader applicability in other domains with networked data.

The performance of the newly proposed techniques is experimentally tested, and a new profit driven evaluation methodology is applied to assess the results of two real life case studies on large scale telco data sets, containing both networked (call detail record data) and non-networked (customer related) information about millions of subscribers. The experiments indicate the existence of a significant impact of social network effects on the churn behavior of telco subscribers. Interestingly, also non-Markovian social network effects are observed: the churn behavior of not only the friends, but also the friends of friends have an impact on the churn behavior of telco subscribers.

Relational and non-relational classifiers, which are built using respectively networked and non-networked information, are found to detect different groups or types of churners, hinting towards a great potential of social network analysis for customer churn prediction. A propositionalization approach to incorporate social network information within a non-relational model by including network variables yields the best performing stand-alone CCP model. However, the novel parallel modeling setup, resulting in a non-integrated model, outperforms all other approaches to combine a relational and non-relational model and generates significant gains in profit compared to integrated models, including the propositionalization approach. Hence, the main finding of this study with strong consequences for the practice of customer relationship management is that applying relational classifiers in combination with the current generation of CCP models can generate significant profit gains. This results from the fact that relational and non-relational CCP models detect different types of churners. Finally, including second order network effects by upgrading the network neighborhood using the weight product, is shown to improve the overall classification power and to further increase the generated profits compared to a model that restricts the impact of the network to first order effects.

Acknowledgements

We extend our gratitude to the Flemish Research Council for financial support (Odysseus grant B.0915.09), and the National Bank of Belgium (NBB/10/006).

Appendix A.

Proof. Proof of Theorem 1 Assuming a weight is the equivalent of an inverted distance, a weight matrix $W = (w_{ij})$ can be transformed into an equivalent distance matrix $D = (d_{ij})$ by taking the inverse of each weight in the matrix, i.e. $(w_{ij}) = 1/(d_{ij})$. Subsequently, the distance matrix (d_{ij}) can be upgraded to the second order distance matrix D^2 by applying the distance product, $d_{ij}^{\star 2} = d_{ij} \star d_{ij}^T = \min_k (d_{ik} + d_{kj}^T)$. Finally, the resulting distance product matrix can be converted in a weights matrix again, $(w_{ij}^{\circledast 2}) = 1/(d_{ij}^{\star 2})$. This sequence of operations can be expressed as a single operation, i.e. the weight product as defined by Definition 9.

Let us first express $d_{ij}^{\star 2}$ in terms of the weight matrix as follows:

$$\begin{aligned} d_{ij}^{\star 2} &\equiv \min_k \left(d_{ik} + d_{kj}^T\right) \\ &= \min_k \left(\frac{1}{w_{ik}} + \frac{1}{w_{kj}^T}\right) \\ &= \min_k \left(\frac{w_{ik} + w_{kj}^T}{w_{ik} \cdot w_{kj}^T}\right). \end{aligned} \tag{18}$$

Next, the distance product matrix (d_{ij}^2) within the equation $(w_{ij}^2) = 1/(d_{ij}^2)$ can be substituted by Eq. (18), which formally proofs Theorem 1:

$$\begin{aligned} w_{ij}^{\circledast 2} &= \frac{1}{d_{ij}^{\star 2}} \\ &= \frac{1}{\min_k(w_{ik} + w_{kj}^T / w_{ik} \cdot w_{kj}^T)} \\ &= \max_k \left(\frac{w_{ik} \cdot w_{kj}^T}{w_{ik} + w_{kj}^T}\right) \\ &= w_{ij} \circledast w_{ij}^T \end{aligned} \tag{19}$$

□

References

[1] B. Baesens, R. Setiono, C. Mues, J. Vanthienen, Using neural network rule extraction and decision tables for credit-risk evaluation, Management Science 49 (3) (2003) 312–329.

[2] P. Blau, Inequality and Heterogeneity: A Primitive Theory of Social Structure, NY Free Press, New York, NY, USA, 1977.

[3] S. Boccaletti, V. Latora, Y. Moreno, M. Chavez, D. Hwang, Complex networks: structure and dynamics, Physics Reports 424 (2006) 175–308.

[4] F. Bonchi, C. Castillo, A. Gionis, A. Jaimes, Social network analysis and mining for business applications, ACM Transactions on Intelligent Systems and Technology 2 (3) (2011) 1–37.

[5] L. Breiman, Bagging predictors, Machine Learning 24 (2) (1996) 123–140.

[6] L. Breiman, Random forests, Machine Learning 45 (1) (2001) 5–32.

[7] W. Buckinx, D. Van den Poel, Customer base analysis: partial defection of behaviourally loyal clients in a non-contractual FMCG retail setting, European Journal of Operational Research 164 (1) (2005) 252–268.

[8] J. Burez, D. Van den Poel, Handling class imbalance in customer churn prediction, Expert Systems with Applications 36 (3) (2009) 4626–4636.

[9] S. Chakrabarti, B. Dom, P. Indyk, Enhanced hypertext categorization using hyperlinks., in: Proceedings of the ACM SIGMOD International Conference on Management of Data, 1998, pp. 307–319.

[10] N. Christakis, J. Fowler, The spread of obesity in a large social network over 32 years, New England Journal of Medicine 357 (4) (2007) 370–379.

[11] D.R. Cox, N. Wermuth, Multivariate Dependencies: Models, Analysis and Interpretation, Vol. 67, Chapman & Hall/CRC, 1996.

[12] K. Dasgupta, R. Singh, B. Viswanathan, D. Chakraborty, S. Mukherjea, A. Nanavati, A. Joshi, Social ties and their relevance to churn in mobile telecom networks., in: Proceedings of the 11th international conference on Extending Database Technology: Advances in database technology, EDBT '08, 2008, pp. 697–711.

[13] S. Džeroski, N. Lavrač, Relational Data Mining, Kluwer, Berlin, Germany, 2001.

[14] Y. Freund, L. Trigg, The alternating decision tree learning algorithm., in: Proceedings of the 16th International Conference on Machine Learning, ICML, 1999, pp. 124–133.

[15] S. Geman, D. Geman, Stochastic relaxation, gibbs distributions and the bayesian restoration of images, IEEE Transactions on Pattern Analysis and Machine Intelligence 6 (1984) 721–741.

[16] L. Getoor, B. Taskar, Statistical Relational Learning, MIT Press, Cambridge, MA, USA, 2007.

[17] S. Gregor, I. Benbasat, Explanations from intelligent systems: theoretical foundations and implications for practice, MIS Quarterly 23 (4) (1999) 497–530.

[18] T. Hastie, R. Tibshirani, J. Friedman, The Elements of Statistical Learning, Data Mining, Inference and Prediction, Springer, New York, NY, U.S.A, 2001.

[19] D. Jensen, J. Neville, Linkage and autocorrelation cause feature selection bias in relational learning., in: Proceedings of the 19th International Conference on Machine Learning, ICML, 2002, pp. 259–266.

[20] D.A. Johannsen, D.J. Marchette, Betti numbers of graphs with an application to anomaly detection, Statistical Analysis and Data Mining 5 (3) (2012) 235–242.

[21] S. Kramer, N. Lavrač, P. Flach, Relational Data Mining, Kluwer, Berlin, Germany, 2001, pp. 262–286, Ch. Propositionalization approaches to relational data mining.

[22] A. Lemmens, C. Croux, Bagging and boosting classification trees to predict churn, Journal of Marketing Research 43 (2) (2006) 276–286.

[23] Q. Lu, L. Getoor, Link-based classification., in: Proceedings of the 20th International Conference on Machine Learning, ICML, 2003, pp. 496–503.

[24] S. Macskassy, F. Provost, Classification in networked data, Journal of Machine Learning Research 8 (2007) 935–983.

[25] D.J. Marchette, Random Graphs for Statistical Pattern Recognition. Vol. 416. Wiley Series in Probability and Statistics, 2005.

[26] D. Martens, F. Provost, Construction and inference of networked data in a bank setting. Working paper CeDER-11-05, Stern School of Business, New York University, 2011.

[27] D. Martens, F. Provost, Explaining data-driven document classifications, MIS Quarterly (2013), in press.

[28] D. Martens, T. Van Gestel, B. Baesens, Decompositional rule extraction from support vector machines by active learning, IEEE Transactions on Knowledge and Data Engineering 21 (2) (2009) 178–191.

[29] M. McPherson, L. Smith-Lovin, J. Cook, Birds of a feather: homophily in social networks, Annual Review of Sociology 27 (2001) 415–444.

[30] A. Nanavati, R. Singh, D. Chakraborty, K. Dasgupta, S. Mukherjea, G. Das, S. Gurumurthy, A. Joshi, Analyzing the structure and evolution of massive telecom graphs, IEEE Transactions on Knowledge and Data Engineering 20 (5) (2008) 703–718.

[31] S. Neslin, S. Gupta, W. Kamakura, J. Lu, C. Mason, Detection defection: measuring and understanding the predictive accuracy of customer churn models, Journal of Marketing Research 43 (2) (2006) 204–211.

[32] J. Neville, D. Jensen, Relational dependency networks, Journal of Machine Learning Research 8 (2007) 653–692.

[33] M. Newman, Networks: An Introduction, Oxford University Press, Oxford, UK, 2010.

[34] C. Perlich, F. Provost, Aggregation-based feature invention and relational concept classes., in: Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD-03, 2003, pp. 167–176.

[35] C. Perlich, F. Provost, Distribution-based aggregation for relational learning with identifier attributes, Machine Learning 62 (1/2) (2006) 65–105.

[36] Y. Richter, E. Yom-Tov, N. Slonim, Predicting customer churn in mobile networks through the analysis of social groups, in: Proceedings of the 10th SIAM International Conference on Data Mining, 2010, pp. 732–741.

[37] J. Rocchio, Relevance feedback in information retrieval, in: G. Salton (Ed.), The SMART Retrieval System—Experiments in Automatic Document Processing, Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1971, pp. 313–323.

[38] K. Subramanian, S. Suresh, A meta-cognitive sequential learning algorithm for neuro-fuzzy inference system, Applied Soft Computing 12 (11) (2012) 3603–3614.

[39] W. Verbeke, K. Dejaeger, D. Martens, J. Hur, B. Baesens, New insights into churn prediction in the telecommunication sector: a profit driven data mining approach, European Journal of Operational Research 218 (1) (2012) 211–229.

[40] W. Verbeke, D. Martens, C. Mues, B. Baesens, Building comprehensible customer churn prediction models with advanced rule induction techniques, Expert Systems with Applications 38 (3) (2011) 2354–2364.

[41] U. Zwick, All pairs shortest paths using bridging sets and rectangular matrix multiplication, Journal of the Association for Computing Machinery 49 (3) (2002) 289–317.