Soc. Netw. Anal. Min. (2016) 6:86
DOI 10.1007/s13278-016-0397-y

ORIGINAL ARTICLE

A supervised approach for intra-/inter-community interaction prediction in dynamic social networks

Giulio Rossetti[1] · **Riccardo Guidotti**[2] · **Ioanna Miliou**[2] · **Dino Pedreschi**[2] · **Fosca Giannotti**[1]

Received: 17 December 2015 / Accepted: 17 September 2016 / Published online: 27 September 2016

Abstract Due to the growing availability of Internet services in the last decade, the interactions between people became more and more easy to establish. For example, we can have an intercontinental job interview, or we can send real-time multimedia content to any friend of us just owning a smartphone. All this kind of human activities generates digital footprints, that describe a complex, rapidly evolving, network structures. In such dynamic scenario, one of the most challenging tasks involves the prediction of future interactions between couples of actors (i.e., users in online social networks, researchers in collaboration networks). In this paper, we approach such problem by leveraging networks dynamics: to this extent, we propose a supervised learning approach which exploits features computed by time-aware forecasts of topological measures calculated between node pairs. Moreover, since real social networks are generally composed by weakly connected modules, we instantiate the interaction prediction problem in two disjoint applicative scenarios: intra-community and inter-community link prediction. Experimental results on real time-stamped networks show how our approach is able to reach high accuracy. Furthermore, we analyze the performances of our methodology when varying the typologies of features, community discovery algorithms and forecast methods.

Keywords Link prediction · Community discovery · Time series

✉ Giulio Rossetti
giulio.rossetti@isti.cnr.it

Riccardo Guidotti
riccardo.guidotti@di.unipi.it

Ioanna Miliou
ioanna.miliou@for.unipi.it

Dino Pedreschi
dino.pedreschi@di.unipi.it

Fosca Giannotti
fosca.giannotti@isti.cnr.it

[1] KDDLab, ISTI -CNR, Via G. Moruzzi,1, 56124 Pisa, Italy

[2] KDDLab, University of Pisa, Largo B. Pontecorvo, 3, 56127 Pisa, Italy

1 Introduction

Complex networks are nowadays used to describe a wide range of real-world phenomena: social and biological interactions, economic systems as well as optimization problems are examples of how broad is becoming the range of topics which are studied using network science approaches. This breadth of applicative scenarios is one of the main reasons for the renewed interest in network analysis that, in recent years, is emerged in the scientific community. Indeed, a wide class of network problems have been analyzed and applied to several branches of research: community discovery, link prediction, node ranking and classification are only few of the several tasks extensively investigated. Among all those tasks, the most challenging and interesting ones aim to describe how networks evolve through time.

Networks are rarely used to model static entities: i.e., in social contexts we can observe that as time goes by users appear and disappear, new interactions take place, and existing ones fell apart disrupting existing paths. Understanding these dynamics is the first step to obtain insights into the real nature of the phenomenon modeled by the observed network. Moreover, almost all the network problems can be reformulated in order to take into account

the temporal dimension: communities can be tracked through all their life cycle to unveil their history; incremental ranking can be computed in order to optimize execution costs; links can be predicted using information obtained by the analysis of topology changes in the local surroundings of nodes. Networks taking into account the temporal dimension are called *dynamic*. The topology of these networks evolves over time as new links and nodes may appear or disappear according to the interactions among their users.

In order to analyze dynamic networks in a reliable way, the social features affecting their structure and behavior must be considered. Indeed, temporal changes are sometimes independent from the network topology itself and result from external factors. The problem of predicting the existence of hidden links or the creation of new ones in social networks is commonly referred to as the *link prediction* problem. In this work, we propose an analytic process which, exploiting well-known state-of-the-art techniques, is able to tackle this challenging task in dynamic networks.

In order to capture how topological features evolve—knowledge needed to perform prediction in dynamic contexts—we made use of time series. Specifically, considering a dynamic social network, we built a time series for each social feature of each couple of nodes, that is a sequence of measures at successive points in time, spaced at uniform time intervals. In our approach, we used such structure to forecast future values of each feature: time series forecasts are then used to solve the link prediction problem.

Several works highlight that, when addressing link prediction through supervised learning, it does not appear to exist a set of features or a similarity index that is outperforming in all settings: depending on the network analyzed, various measures could be particularly promising or not (Liben-Nowell and Kleinberg 2007). This suggests that the predictors which work best for a given network may be related to the structure within the network rather than a universal best set of predictors. Topological similarity indexes encode information about the relative overlap between nodes' neighborhoods. We expect that the more similar two nodes' neighborhoods are (e.g., the more overlap in shared friends), the more likely they may be to exhibit a future link. Moreover, we exploit well-known social network characteristics such as power law degree distribution (Barabási and Albert 1999), the small-world phenomenon (Watts and Strogatz 1998), and community structure (Girvan and Newman 2002).

In this study, a valuable topological information that we leverage regards the modular structure of social networks: indeed, social networks can be partitioned into densely and internally connected vertex sets and it has been extensively observed that such topologies provide bounds to the sociality of the users within them. Furthermore, in a dynamic scenario, more than in a static one, the evolution of such boundaries describes changes in people' social behaviors. Starting from such observation, we decided to divide the original problem into two disjoint tasks:

- *intra-community* interaction prediction;
- *inter-community* interaction prediction.

Following the hypothesis that friends of friends are more likely to become friends than individuals who have no friends in common (Granovetter 1973; Rapoport 1963), in the former task we restrict our attention to the prediction of new links at time $t+1$ which occur between individuals who are in the same community at least once in $[0, t]$. This strategy has the computationally not negligible advantage of calculating only the features among nodes belonging to the same community. The latter task, on the other hand, focuses on the forecast of future bridges across network modules: such interactions represent the weak ties that keep together the overall network structure.

In this paper, we propose a data mining process able to provide a solution to both tasks: moreover, we formalize the link prediction problem for dynamic networks, the *Interaction Prediction*. Our approach predicts future interactions by combining dynamic social networks analysis, time series forecast, feature selection and network community structure.

The rest of this paper is organized as follows. In Sect. 2 is reported the formal definition of the link prediction problem studied. Section 3 illustrates the detail of the proposed approach as a workflow. In Sect. 4 are reported the experimental results, for both *intra-community* and *inter-community* interaction prediction tasks, obtained using real-world datasets. Section 5 introduces the related works for the link prediction problem. Finally, in Sect. 6 conclusions and future works are summarized.

2 Interaction prediction problem

The classic formulation of *link prediction* involves the use of the observed network status to predict new edges that are likely to appear in the future or to unveil hidden connections among existing nodes. To satisfy this definition, a wide set of approaches were proposed and tested on several different domains both in supervised and in unsupervised fashion. Graph structures are often used to describe rapid-scale human dynamics: social interactions, call graphs, buyer–seller scenarios and scientific collaborations are only few examples. This is exactly the reason why *link prediction* has become the principal instrument used to

address the need of dealing with networks that evolve through time.

In this work, our aim is to exploit the temporal information carried by the appearance and disappearance of edges in a fully dynamic context: doing so, we plan to overcome the limitations imposed by the analysis of a static scenario when making predictions. To model rapid-scale dynamics, we will adopt the *interaction network* model:

Definition 1 (*Interaction Network*) An *interaction network* $G = (V, E, T)$, is defined by a set of nodes V and a set of time-stamped edges $E \subseteq V \times V \times T$ describing the interactions among them. An edge $e \in E$ is thus described by the triple (u, v, t) where $u, v \in V$ and $t \in T$. Each edge e represents an interaction between nodes u and v that took place at time t.

To easily analyze an interaction network G, we discretize it into τ consecutive snapshots of the same duration, thus obtaining a set of graphs $\mathcal{G} = \{G_0 \ldots, G_\tau\}$. We assume that the interactions belonging to G_t are only the ones that appear in the interval $(t, t+1)$. Such modeling choice allows us to make predictions not only for interactions that will take place among previously unconnected nodes, but also for predicting edges that have already appeared in the past. This decision is made in order to better simulate the dynamics that real interaction networks exhibit allowing nodes and edges both to rise and to fall. In real interaction networks, this model is a good proxy for structural dynamics since it allows to implicitly assign a time to leave to links (i.e., in a call graph, it enables to weight more recent interactions w.r.t. older ones when predicting future contacts among a pair of nodes). Due to the adoption of this more complex graph model, hereafter we will refer to this peculiar formulation of the LP problem as *Interaction Prediction* problem:

Definition 2 *(Interaction Prediction)* Given a set $\mathcal{G} = \{G_0, \ldots G_t, \ldots G_\tau\}$ of ordered network observations, with $t \in T = \{0 \ldots \tau\}$, the *interaction prediction* problem aims to predict new interactions that will took place at time $\tau + 1$ thus composing $G_{\tau+1}$.

In the following section, we introduce our analytical workflow, built upon a supervised learning strategy, designed to solve the Interaction Prediction problem.

3 Proposed approach

The Interaction Prediction problem introduces new challenges to an already complex task. Due to the evolutionary behavior of the networks subject of our investigation, a particular effort is needed in order to find a reasonable way to take care of structural dynamics during the prediction phase. To this extent, we make use of time-stamped network observations and community knowledge besides classical features in order to learn a robust machine learning model able to forecast new interactions. We design our approach to follow four steps (graphically represented in Fig. 1):

Step 1 Given an interaction network G as input, for each temporal snapshot $t \in T$ we compute a partition $\mathcal{C}_t = \{C_{t,0}, \ldots, C_{t,k}\}$ of G_t using a community discovery algorithm. Then we define, for each t and C, $G_{C_t} = (V_{t,C}, E_{t,C})$ as the subgraph induced on G_t by the nodes in C_t, such that $V_{t,C} \subseteq V_t$ and $E_{t,C} \subseteq E_t$.

Step 2 For each $t \in T$, we consider the interaction communities $\mathcal{C}_t$ of G_t and compute a set of measures F for each pair of nodes pair $(u, v) \in W_{t,\mathcal{C}}$ such that $W_{t,\mathcal{C}} = \{(u, v) : u, v \in V_{t,C} \wedge C_t \in \mathcal{C}_t\}$, that is (u, v) belong to the same community at time t. Thus, we obtain values $f_t^{u,v}$ describing *structural* features, *topological* features and *community* features of the node pairs (u, v) at time t.

Step 3 With these values, for each couple of nodes $(u, v) \in W_{t,\mathcal{C}}$ and feature $f \in F$ we build a time series $S_f^{u,v}$ using the sequence of measures $f_0^{u,v}, f_1^{u,v}, \ldots, f_\tau^{u,v}$. Then, we apply well-known forecasting techniques in order to obtain its future expected value $f_{\tau+1}^{u,v}$.

Step 4 Finally, we use the set of expected values $f_{\tau+1}^{u,v}$ for each feature $f \in F$ to build a classifier that will be able to predict future intra-community interactions.

In the following, we discuss each step by itself, proposing solutions that can be used to instantiate the described analytical process making use of well-known methodologies.

3.1 Step 1: community discovery

Partitioning a network into communities is a complex task: for this reason, several approaches were introduced during the last decade, each one of them tailored to extract communities carrying specific traits. Due to the absence of an universally shared community definition, in order to evaluate the impact of community structure on the predictive power of the proposed supervised learning strategy, we tested three different CD algorithms, namely *Louvain*, *Infohiermap* and *DEMON*. Here we provide a short description of their major characteristics, while in the experimental section we will discuss how they affect the predictive power of the described analytical process. We

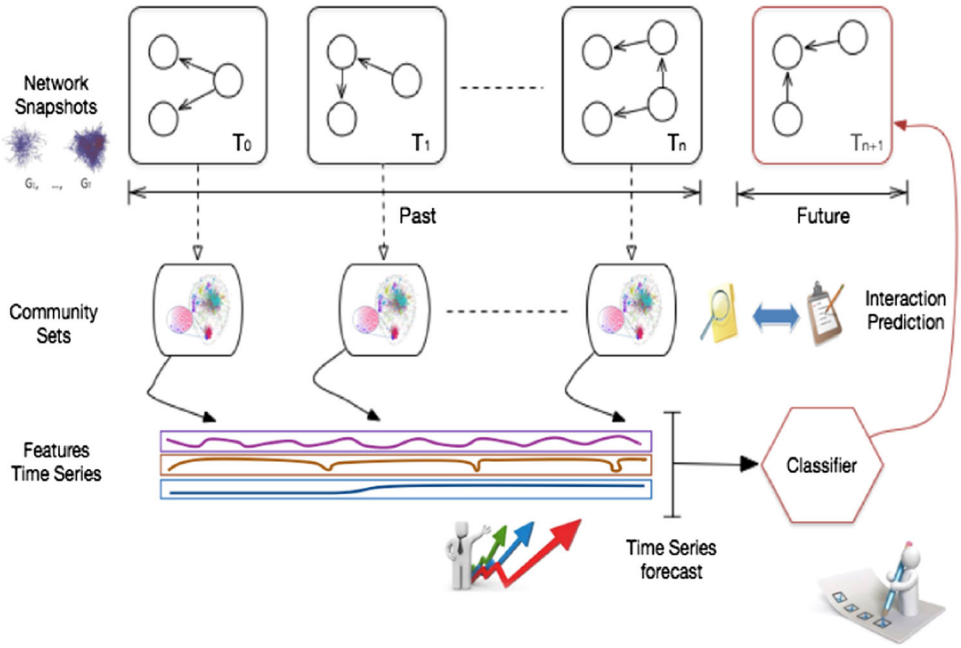

Fig. 1 Proposed approach workflow. The interaction network is split into network snapshots and each snapshot is partitioned using a community discovery algorithm (*Step 1*). Then for each community, a large set of features describing nodes and links are calculated (*Step 2*). Using these values, different time series are built and a forecast of their future values is provided for the time of the prediction (*Step 3*). Finally, these expected values are used to train a classifier able to predict new interactions (*Step 4*)

remind that we adopted community discovery algorithms to split interaction networks into communities, and then we used these communities to calculate the features that will be illustrated in the following and to perform the predictions of new interactions.

Louvain is an heuristic method based on modularity optimization (Blondel et al. 2008). It is fast and scalable on very large networks and reaches high accuracy on ad hoc modular networks. The optimization is performed in two steps. First, it looks for "small" communities by optimizing modularity locally. Second, it aggregates nodes belonging to the same community and builds a new network whose nodes are the communities. These steps are repeated iteratively until a maximum of modularity is attained and a hierarchy of communities is produced. Louvain produces a complete non-overlapping partitioning of the graph. As most of the approaches based on modularity optimization, it suffers from a "scale" problem that causes the extraction of few big communities and a high number of very small ones.

Infohiermap is one of the most accurate and best performing hierarchical non-overlapping clustering algorithms for community discovery (Rosvall and Bergstrom 2011) studied to optimize community conductance. The graph structure is explored with a number of random walks of a given length and with a given probability of jumping into a random node. Intuitively, the random walkers are trapped in a community and exit from it very rarely. Each walk is described as a sequence of steps inside a community followed by a jump. By using unique names for communities and reusing a short code for nodes inside the community, the walk description can be highly compressed, in the same way as reusing street names (nodes) inside different cities (communities). The renaming is done by assigning a Huffman coding to the nodes of the network. The best network partition will result in the shortest description for all the walks.

DEMON is an incremental and limited time complexity algorithm for community discovery (Coscia et al. 2012). It extracts ego networks, i.e., the set of nodes connected to an ego node u, and identifies the real communities by adopting a democratic bottom-up merging approach of such structures. Following this approach, each node, through its ego network (i.e., the induced graph on his one-hop neighborhood), gives the perspective of the communities surrounding it: all the different nodes perspectives are then merged together leading to an overlapping partition. To each ego network is applied a label propagation algorithm which ignores the presence of the ego itself in order to identify local micro-communities, and then, with equity, such individual micro-level is combined with the ones obtained by the rest of the nodes ego networks. The result of this combination is a set of overlapping modules, the guess of the real communities in the global system, made not by an external observer, but by the actors of the network itself.

We chose to use the aforementioned algorithms since, due to their formulations, they cover three different kinds of community definitions: modularity-, conductance- and density-based ones. Since in our test we vary the structural properties of the communities used to extract the classification features, in the experimental analysis we will be able to discuss which network partitioning approach is able to provide more useful insights into future interactions.

3.2 Step 2: features design

In order to efficiently approach the Interaction Prediction task using a supervised learning strategy, it is crucial to identify and calculate a valuable set of features to train the classifier. When dealing with large-scale graphs that may include millions of vertices and links, one of the challenges is the computationally intensive extraction of such features. Several studies related to link prediction such as Feng et al. (2012), Fire et al. (2013), Jahanbakhsh et al. (2012), Lichtnwalter and Chawla (2012), Xu and Rockmore (2012) have tried to suggest which are the optimal topological structure of a network and the best features to be used. Moving from the results of such analysis, we decided to use information belonging to three different families: *pairwise structural features*, *global topological features* and *community features*. We recall that all the features were computed before the community extraction phase on node pairs sharing the same social context.

3.2.1 Pairwise structural features

In this class fall all the measures used in the literature to score the likelihood of new links in unsupervised scenarios. Starting from the measures proposed in Liben-Nowell and Kleinberg (2007), we restricted our set to the one in Table 1.

Given a graph G, we will use the following notation: $\Gamma(u)$ identifies the set of neighbors of a node u in G; $|\bullet|$ represents the cardinality of the set $\bullet$.

- *Common Neighbor* (*CN*) assigns as likelihood score of a new link the number of neighbors shared by endpoints (Newman 2001).
- *Jaccard Coefficient* (*JC*) measures the likelihood of two nodes to establish a new connection as the ratio among their shared neighbors and the total number of their distinct neighbors (Salton and McGill 1983).
- *Adamic Adar* (*AA*) refines CN by increasing the importance of nodes which possess less connections (Adamic and Adar 2003).
- *Preferential Attachment* (*PA*) assumes that the probability of a future link between two nodes is proportional to their degree (Barabási and Albert 1999).

As a direct consequence to their formulation, CN, JC and AA share the same result set composed by all the pair of nodes at most two-hops in G. However, the values obtained by the three measures for the same edge do not correlate (i.e., having a high CN does not imply having high JC or AA). Conversely, PA generates scores for all the possible node pairs: we restrict its computation to nodes at most at distance two to uniform its result set to the ones of the other measures. We remind that in our calculus of the features G corresponds to G_{C_t}, that is the subgraphs induced on G_t for each time stamp t.

3.2.2 Global topological features

The features discussed so far look at the nodes immediate surroundings. However, also the position of a node within the network carries valuable information that can be exploited in order to predict which kind of nodes are attracted by it.

In the literature, a wide set of measures were proposed to estimate the centrality of nodes and edges as well as their rank within a network. These scores are, often, computationally expensive to calculate: for this reason we have decided to make use only of two of them whose definition is reported in Table 2 and calculated on G_{C_t}.

- *Degree Centrality* (*DC*) relates the centrality of a node to its degree.
- *PageRank* (*PR*) is a link analysis algorithm introduced by Page et al. (1999) and used by the *Google* Web search engine. It assigns a numerical score to each element of a hyperlinked set of documents with the purpose of measuring its relative importance within the set.

DC and PR scores were computed for both the endpoints of possible edges pairs: the underlying idea is to understand if there is some correlation among the centrality of two nodes and the likelihood of the appearance of a new interaction between them. This choice can be seen as a way to generalize the PA measure where the operator defining the combination of the individual scores is not fixed. For PR,

Table 1 Pairwise structural features

Measure	Description
Common Neighbors (Newman 2001)	$\mathrm{CN}(u,v) = \lvert\Gamma(u) \cap \Gamma(v)\rvert$
Jaccard Coefficient (Salton and McGill 1983)	$\mathrm{JC}(u,v) = \frac{\lvert\Gamma(u)\cap\Gamma(v)\rvert}{\lvert\Gamma(u)\cup\Gamma(v)\rvert}$
Adamic Adar (Adamic and Adar 2003)	$\mathrm{AA}(u,v) = \sum_{w\in\Gamma(u)\cap\Gamma(v)} \frac{1}{\log\lvert\Gamma(w)\rvert}$
Preferential Attachment (Barabási and Albert 1999)	$\mathrm{PA}(u,v) = \lvert\Gamma(u)\rvert \times \lvert\Gamma(v)\rvert$

This kind of features, generally used in unsupervised link prediction, captures the likelihood that a new interaction will happen between a couple of nodes u and v based on their neighbors

Table 2 Global topological features

Measure	Description
Degree Centrality	$\mathrm{DC}(u) = \lvert\Gamma(u)\rvert$
Page Rank (Page et al. 1999)	$\mathrm{PR}(u) = \frac{1-d}{N} + d\sum_{(u,v)\in E}\frac{\mathrm{PR}(v)}{\lvert\Gamma(v)\rvert}$

This set of features model the probability of jumping into a particular node. $\mathrm{PR}(u)$ is the page rank score of node u, N is the total number of nodes, and d is the damping factor. In our experimentation, we used the default value for d (0.85)

we use as dumping factor (d in the formula) its default value (0.85).

3.2.3 Community features

One of the most pressing issues related to LP regards the reduction of *false-positive* forecasts. To this extent, as briefly mentioned before, we exploit community discovery as a way to reduce the number of predictions provided by the chosen pairwise structural features.

Communities group together nodes that are tightly connected within each other than with the rest of the network. Making predictions only between nodes belonging to the same community allows the predictive process to focus on connections that are more likely to appear, thus discarding the ones connecting different graph substructures. However, following the general intuition behind the idea of community, we can take advantage of more specifically designed measures. Indeed, all the information we can gather from the topological analysis of the communities can be used as features describing the extended surroundings of nodes. With this aim, we introduce the set of features summarized in Table 3.

- *Community Size* (*CS*) number of nodes belonging to the community C.
- *Community Edges* (*CE*) number of edges within nodes in C.
- *Shared Communities* (*SC*) identifies the number of communities shared by a couple of nodes. When dealing with network partitions, *SC* takes value in $\{0, 1\}$, while in case of overlapping communities its domain is $[0, |\mathcal{C}|]$.
- *Community Density* (*D*) ratio of edges belonging to the community over the number of possible edges among all the nodes within it.
- *Transitivity* (*T*) identifies the ratio of triangles with respect to open "triads" (two edges with a shared vertex).
- *Max Degree* (*MD*) identifies the degree (w.r.t. the community subgraph) of the principal hub for the community.
- *Average Degree* (*AD*) identifies the average degree (w.r.t. the community subgraph) of the nodes within the community.

Table 3 Community features

Measure	Description
Community Size	$CE(G_C) = \lvert E_C\rvert$
Community Edges	$CE(G_C) = \lvert E_C\rvert$
Shared Communities	$CS(u, v, \mathcal{C}) = \lvert\{C \lvert u \in V_C \wedge v \in V_C\ \forall C \in \mathcal{C}\}\rvert$
Community Density	$D(C) = \frac{\lvert E_C\rvert}{\lvert V_C\rvert\times(\lvert V_C\rvert-1)}$
Transitivity	$T = 3\,\frac{\lvert triangles(G_C)\rvert}{\lvert triads(G_C)\rvert}$
Max Degree	$MD(C) = max\{\lvert\Gamma(u)\rvert : u \in V_C\}$
Average Degree	$AD(C) = \frac{\sum_{u\in V_C}\lvert\Gamma(u)\rvert}{\lvert V_C\rvert}$

These features, which are one of the novel contribution of this work, express the relevance of a node in a community

3.3 Step 3: forecasting models

The third step of our approach involves the adoption of time series forecasting models to obtain, given subsequent observation of the same feature for the same pair of nodes, an estimation of its future value. Since the behavior of the observed time series is not known in advance, we adopt several forecasting models based on different underlying assumptions. This choice allows us to identify which one best describes the evolution of the network analyzed later on. Since the time series we are analyzing are not large, we have decided to not employ complex models that are known to be very efficient on extended observation periods. In fact, we tested four computationally efficient models that have shown to achieve good performances on short time series.

In Table 4, we summarize the forecasting approaches tested: in our definitions we identify with $Z_t = (t = 1 \ldots \tau)$ a time series with τ observations and with Θ_t its forecast at time t.

- *Last Value* (*Lv*) considers as forecast the last observed value of the time series.
- *Average* (*Av*) is the average of all the observations in Z_t.
- *Moving Average* (Ma) predicts the next value by taking the mean of the n most recent observed values of a

Table 4 Time series forecasting approaches

Measure	Description
Last Value (Lv)	$\Theta_t = Z_{t-1}$
Average (Av)	$\Theta_t = \frac{\sum_{i=1}^{\tau} Z_i}{\tau}$
Moving Average (Ma)	$\Theta_t = \frac{\sum_{i=\tau-n}^{\tau} Z_i}{n}$
Linear Regression (LR)	$\Theta_{t+h} = \alpha_t + h\beta_t$

These simple methods which are very effective on short time series forecast the future value of a sequence

series Z_t. In our experiments, we have ranged n in the interval $[1, \tau]$.

- *Linear Regression* (LR) fits the time series to a straight line. The level α and the trend β parameter (used to estimate the slope of the line) were defined by minimizing the sum of squared errors between the observed values of the series and the expected ones estimated by the model.

3.4 Step 4: classifier models

Predicting correctly new interactions is not an easy task. The complexity is mainly due to the highly unbalanced class distribution that characterizes the solution space: real-world networks are generally sparse; thus, the number of new interactions over the total possible ones tends to be small. We have discussed how it is possible, at least to some extent, to mitigate this problem by restricting the prediction set (i.e., predicting only new edges among nodes that, during the network history, were involved at least in a common community).

However, even adopting such precautions we can expect a substantial unevenness between the positive and the negative classes. This translates into a very high, hard-to-improve, threshold for the baseline model (i.e., in case of a network having density 0.1, which identifies the presence of "only" 1 / 10 of the possible edges, the majority classifier is capable of reaching more than 0.9 of accuracy by simply predicting the absence of new interactions) even though no interactions will be actually predicted since every possible future links will be marked as not present).

In order to better characterize our approach, we instantiated it in two different scenarios (both for *inter*- and for *intra*-community predictions):

- *Balanced class distribution* we adopted class balancing through downsampling [as performed in previous works (Lichtenwalter et al. 2010)], thus obtaining balanced classes and a baseline model having 0.5 accuracy.
- *Unbalanced class distribution* in order to provide an estimate of the real predictive power expressed by our methodology, we tested it against the unbalanced class distribution as expressed by the original data.

Moreover, since the main focus of this work is to describe a data mining approach that can be used to solve the Interaction Prediction problem and not to discuss a specific classification model, we evaluated our strategy independently from a hosted classifier: for this reason, in the following section we will discuss results achieved by an ensemble of classifiers showing the scores only for the best performing ones. In detail, our supervised learning model set is composed by: decision tree (C4.5, C&R, CHAID, QUEST, random forest), neural network, SVM and logistic regression.

4 Experiments and results

In this section, we report the results obtained by applying our approach to two real-world interaction networks. In Sect. 4.1, the datasets used to perform the experiments are briefly introduced. In Sect. 4.2 are discussed the results obtained focusing the prediction on intra-community interactions: in such context both balanced and unbalanced class scenarios are proposed and used to evaluate our approach. Finally, in Sect. 4.3 the same approach is applied to the forecast of inter-community interactions, the weak links that keep together the modular structure composing complex networks.

4.1 Datasets

We tested our approach on two networks: an interaction network obtained from a Facebook-like[1] *Social* network and a co-authorship graph extracted from *DBLP*[2]. These datasets allow us to test our procedure on two different grounds: a "virtual" context, in which people share thoughts and opinions via a social media platform, and a "professional" one. The general statistics of the datasets are shown in Table 5, while a brief resume is in the following:

Social The Facebook-like social network originates from an online community for students at University of California, Irvine. The dataset includes the users that sent or received at least one message during 6 months. We discretize the network in 6 monthly snapshot and use the first 5 to compute the features needed to predict the edges present in the last one.

DBLP We extract author–author relationships if two authors collaborated at least in one paper. The co-authorship relations fall in temporal window of 10 years (2001–2010). The network is discretized on yearly basis: we use the first 9 years to compute the features and set as target for the prediction the edges belonging to the last one.

In Table 5 we can observe the low average density μ_D of the studied networks across the various snapshots. We notice immediately how the low standard deviation σ_D and σ_{CC} guarantee the good approximation of the average density and clustering coefficient as statistic.

For this reason, it is remarkable the fact that Social is more dense than DBLP even though its clustering

[1] http://toreopsahl.com/datasets/.

[2] http://dblp.org.

Table 5 Networks statistics: average density μ_D, average clustering coefficient μ_{CC} and their standard deviations, σ_D and σ_{CC} reported as representative aggregate among the various snapshot

Network	Nodes	Interactions	#Snapshots	μ_{CC}	σ_{CC}	μ_D	σ_D
DBLP	747,700	5,319,654	10 (years)	0.665	0.018	3.113e−05	9.602e−06
Social	1899	113,145	6 (months)	0.105	0.015	8.600e−03	1.400e−03

We can observe how DBLP is more "partitioning prone" due to the high clustering coefficient. On the other hand, Social has denser snapshots

coefficient is considerably lower than DBLP. This means that, due to its nature, when a new interaction appears in DBLP, more than a couple of users is involved, creating automatically a complete clique, while, in Social, a new interaction just expresses the exchange of a direct message between the two users.

4.2 Intra-community interaction prediction

The Interaction Prediction problem is computationally expensive to address since, in theory, a prediction should be outputted for each pair of nodes in the network analyzed. However, social network are known to be sparse and easily to be partitioned in internally dense substructures. Leveraging this observation, our approach is designed to reduce the node pairs for which compute a prediction to the ones whose endpoints share at least one community membership. Operating this choice, we focus on analyzing strong ties—the links inter-communities—and discard the bridges that connects different communities.

4.2.1 Balanced scenario

It happens frequently, in the LP problem, that the two classes to be predicted, i.e., there will be a link or not, are highly unbalanced. In our case, we have highly unbalanced dataset with a proportion of unlinked–linked of 95.95–4.055 % for Social, and of 98.13–1.87 for DBLP. Unfortunately, the classifiers used in our experiments need a balanced test set in order to build the predictive model in the proper way. Following what is generally done in the literature, we balanced every snapshot G_t for Social and DBLP.

To evaluate the performances of the classifiers, we used the *accuracy* and *AUC* which are defined in terms of the confusion matrix of a binary classifier (see Table 6):

- *Accuracy*, defined as $\text{ACC} = \frac{\text{TP}+\text{TN}}{\text{TP}+\text{FN}+\text{TN}+\text{FP}}$, measures the ratio of correct prediction over the total;
- **AUC** identifies the area under the receiver operating characteristic (ROC). It illustrates the performances of binary classifiers relating the true-positive rate $\text{TPR} = \frac{\text{TP}}{\text{TP}+\text{FN}}$ to the false-positive rate $\text{FPR} = \frac{\text{FP}}{\text{FP}+\text{TN}}$ and providing a visual interpretation useful to compare different models.

To better highlight how the proposed approach performs on real-world networks, we need to compare the outcome of its instantiations varying the combination of community discovery algorithms and time series forecast models used.

We carried out a preliminary study aimed at identifying the optimal window size n for the moving average (Ma) forecast having fixed the community discovery algorithm. By definition, the Lv and Av are special cases of the more general Ma: particularly, the former is equivalent to Ma when $n = 1$, while the latter when $n = \tau$. In Fig. 2 is shown, for the three community discovery algorithms, how the classification accuracy behaves varying the observation window n. We can observe different trends for Social and DBLP networks. In the former, the AUC is maximized by the classifier built upon DEMON communities, while in the latter the same approach is the one with worst performances. This is probably due to the particular definition of ego-network-based overlapping communities provided by this approach which is tailored explicitly for social contexts. Furthermore, by observing these plots we can conclude that, in order to obtain higher performances using Ma, two strategies are consistent: (1) minimize n using as forecast the last value (Lv) in order to make inference approximating the future with the actual network status, or (2) use $n \simeq \tau$ in order to have a better estimation of the whole historical trends. Hereafter, we make use of the best scoring classifiers in Fig. 2 to detail our analysis. We will refer to them as the Ma models for each specific network and community definition.

As second step, we compare the outcomes of the classifiers built using the LR forecast models with the Ma ones. In Fig. 3 are shown the ROC curves for both Social and

Table 6 Confusion matrix of a binary classifier

	Predicted	
	Class 0	Class 1
Actual		
Class 0	TN (true neg.)	FP (false pos.)
Class 1	FN (false neg.)	TP (true pos.)

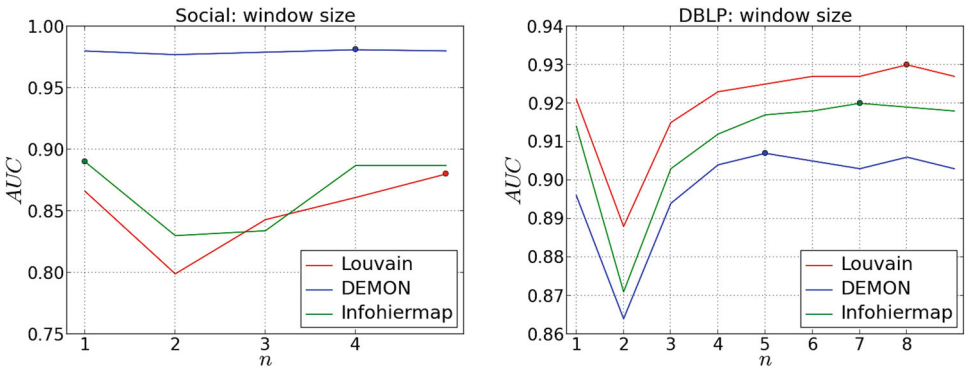

Fig. 2 Balanced scenario. Accuracy *AUC* behavior varying the observation window $n \in [0, \tau]$ using the Moving Average Ma. *Dots highlight* highest values

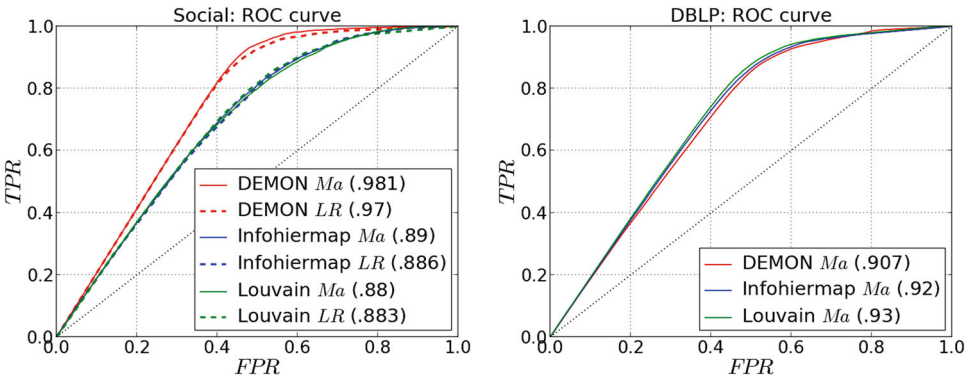

Fig. 3 Balanced scenario. ROC curves of the various proposed workflow executed with different community discovery algorithms and forecasting methods. In Social the best performer is DEMON with Moving Average, while in DBLP there is not a combination considerably better than the others

DBLP datasets. In the former network, we can observe how LR and Ma provide very similar results even if the moving average is always capable of obtaining slightly better performances. DBLP shows the same trend with a small gap between the two approaches (for this reason, we omit the LR curve). We report in Table 7 the AUC and the ACC for all the comparisons.

Once identified the two best performers for Social (DEMON Ma and Infohiermap Ma) and for DBLP (Louvain Ma and Infohiermap Ma) w.r.t. AUC and ACC, we investigated which are the key features that contribute to their performances. We report in Fig. 4 the relative importance of the features used by the classifiers for each method. We can see how in Social the classifier built upon DEMON *(a)*, as well as the one using Infohiermap communities *(c)*, gives high importance to degree centrality and community measures (in particular to density, size and average degree) and tends to make less discriminating decision using pairwise structural features (with the exception of PA). Conversely, in DBLP (*b, d, e*) the community features set seems to show small predictive power for both the analyzed algorithms. This discrepancy is probably due to the different nature of the studied networks: Social naturally models real social interactions in a short period, while DBLP is inferred from connections (working collaborations) that are developed through years.

Table 7 Balanced scenario

Network	*DBLP*		*Social*	
Algorithm	AUC	ACC (%)	AUC	ACC (%)
DEMON Ma	0.907	85.58	**0.981**	93.55
DEMON LR	0.901	84.35	0.970	91.87
Louvain Ma	**0.930**	87.72	0.880	80.27
Louvain LR	0.926	87.48	0.883	81.37
Infohiermap Ma	0.920	86.69	0.890	81.34
Infohiermap LR	0.917	86.18	0.886	80.89

Compared performances varying community discovery and forecasting methods. In bold are the best performers. We can observe how the prediction is more method independent in DBLP than in Social

In order to understand the boost provided to the classifier by the adoption of the right community discovery algorithm, we designed two different baselines: Structural Forecast (*SF*) and Filtered Structural Forecast (*FSF*). The SF model trains the classifier using only the forecasts for the pairwise structural features (CN, AA, PA and JC) computed on all the couple of nodes at distance at most 3 hops present in the whole network, not taking into account the presence/absence of shared communities among them. On the other hand, the FSF model restricts the computation to the pair of nodes belonging to the same community as the proposed approach does. As case study we report in

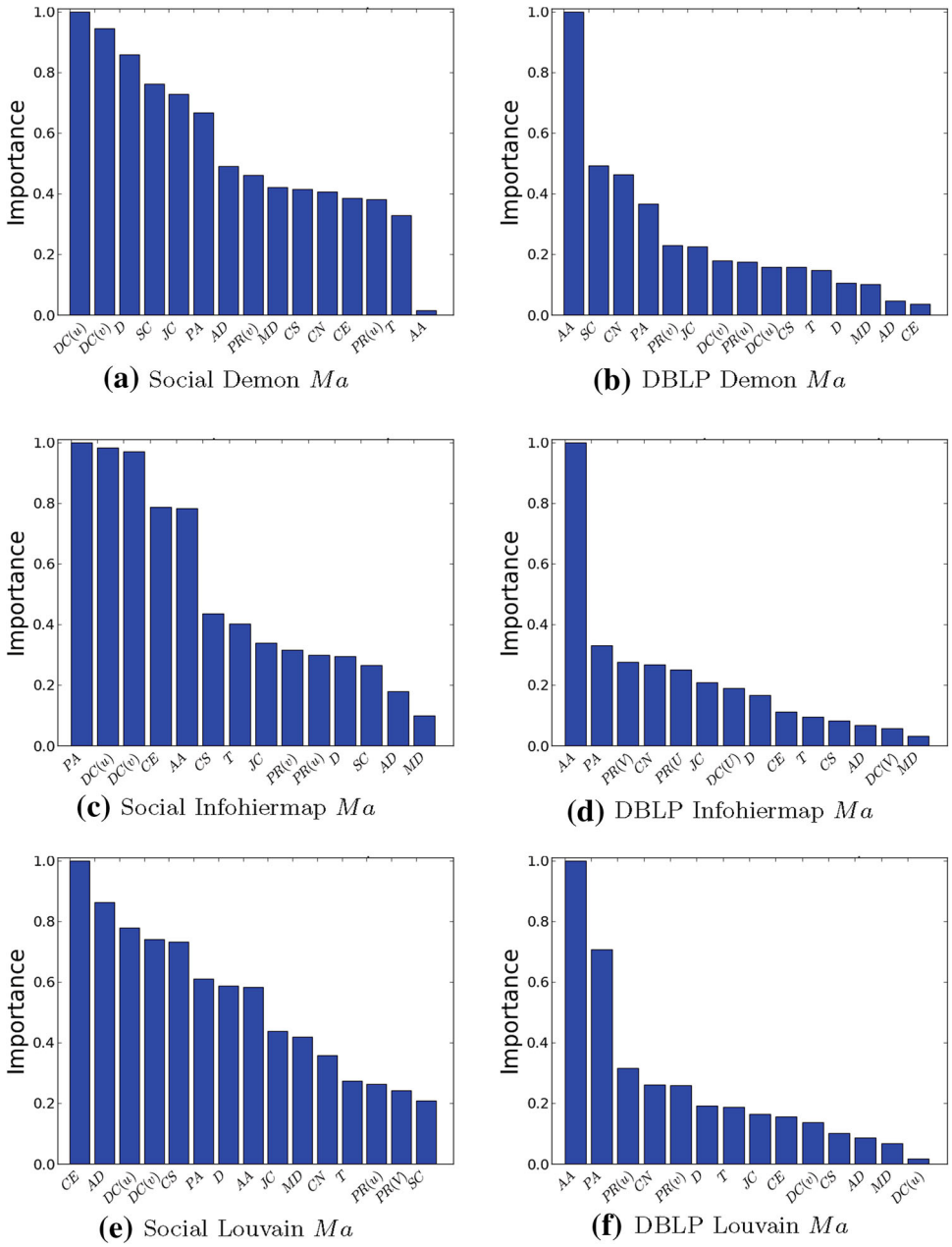

Fig. 4 Balanced scenario. Features importance: the classifiers built for Social (in particular **a** and **c**) give high importance to community average degree DC, density *D* and size *SC*. On the other hand, for DBLP the most important features are the Adamic Adar AA and preferential attachment PA

Table 8 AUC and ACC of the best Ma and LR baselines for the Social dataset.

Since in Social our best performing approach is the one built upon DEMON communities, the structural features for the FSF baseline were computed using such partition of the network. The obtained results show that, using features extracted from the communities, we are able to gain 0.025 in AUC and 3.45 % in ACC with respect to the FSF Ma baseline, and 0.08 in AUC and 10.67 % in ACC with respect to the FS Ma one. These results highlight the importance of communities for the interaction prediction task, not only in providing features for pair of nodes, but also in filtering the dataset in order to determine a more accurate selection of nodes for the prediction. Without loss of generality, in the rest of this section, in order to reduce the number of comparisons, we will report a full analysis only for the Social dataset. Furthermore, the results obtained for the DBLP scenario do not differ significantly from the ones discussed with the exception, as seen previously, of the best community discovery algorithm (Louvain instead of DEMON). This divergence is due to the different nature and topology of the networks analyzed.

Feature Class Prevalence Since our models are built upon three different classes of features (structural, topological and community related), it is mandatory to test their results against the classifiers using them separately.

Table 8 Balanced scenario (social)

Algorithm	AUC	ACC (%)
SF Ma	0.901	82.88
SF LR	0.895	82.18
FSF Ma	**0.956**	90.10
FSF LR	0.937	88.09

Baselines on structural features using only Structural Forecast (*SF*) features calculated in the whole network and Filtered Structural Forecast (*FSF*) calculated following the proposed approach

In bold the AUC of the best performing approach

Such analysis allows us to assess the predictive power of each class of features, giving an idea of their overall importance for the complete model. We built a classifier for each community discovery algorithm and each feature class by using together all the forecasted versions of the features belonging to it. As shown in Table 9, regardless of the community discovery algorithm used, the most predictive features are the ones belonging to the *topology* class, followed by *structural* and *community* ones. However, we can observe how the AUC and ACC are always higher for the model based on the DEMON approach: this trend suggests that this algorithm is the one that better bounds, at least for this network, the nodes that are more likely to establish future interactions.

Complete Classifier We investigated if the performances of the analyzed classifiers can be improved by combining all the features obtained at the end of the forecasting stage (i.e., all the time series forecasts computed with Ma and LR). As we can see in Table 10, the performance boost is negligible with respect to DEMON Ma; in fact, we are able to gain only 0.35 % in ACC maintaining the same AUC

Table 9 Balances scenario (social)

Algorithm	AUC	ACC (%)
DEMON *Structural*	0.957	90.59
DEMON *Topology*	0.962	91.44
DEMON *Community*	0.903	83.53
Louvain *Structural*	0.850	78.63
Louvain *Topology*	0.875	79.38
Louvain *Community*	0.724	66.64
Infohiermap *Structural*	0.876	79.85
Infohiermap *Topology*	0.887	80.81
Infohiermap *Community*	0.667	62.11

Compared classifier performances for different classes of features. We can notice how independently from the community discovery algorithm the topological features always provide the highest performances

Table 10 Balanced scenario (social)

Algorithm	AUC	ACC (%)
DEMON *All*	**0.981**	93.90
Louvain *All*	0.901	83.05
Infohiermap *All*	0.894	81.91
FS *All*	0.959	90.44

Compared classifier performances using all the features. DEMON reaches the highest performances in terms of accuracy and area under the curve

In bold the AUC of the best performing approach

w.r.t. the results shown in Table 7. This means that the feature set used by our best classifier is "stable": its extension does not produce advantages that justify an increase of model complexity. Conversely, for Louvain and Infohiermap the gain in AUC and ACC is more evident: this is due to the different degree of approximation introduced for each feature in the forecasting stage.

Features forecast correlation As a consequence to the minor deviations in performances for different forecasting methods, we investigated which are the correlations among the forecasted values calculated by LR and Ma with $n \in [0, \tau]$. We analyzed each feature separately observing the correlation average, median and variance. In Table 11, we report the average of the variances of these values aggregated for different classes of features. From this table emerges that, regarding structural features, Louvain has the lowest average of variances of correlations, while, for topological and community related features, it is DEMON with the lowest correlations.

As a result, we can say that, if we use Infohiermap (that has the highest average of the variances) to extract the communities from the interaction network, we should focus on the choice of the different forecasting methods applied. On the other hand, if we calculate the communities with DEMON, it does not matter very much which kind of forecast technique (LR or Ma) we use to calculate the expected values. This statement holds less strongly for Louvain which has a low correlation variance only for structural features.

Table 11 Balanced scenario (social)

Algorithm	Structural	Topology	Community
DEMON	0.023	0.001	0.003
Louvain	0.009	0.017	0.018
Infohiermap	0.042	0.015	0.081

Mean of the variance of the correlations among the values forecasted with LR and *Mv*. The higher this value the most careful must be the choice in selecting the forecasting method

Table 12 Balanced scenario (social)

Algorithm	AUC	ACC (%)
DEMON	0.987	95.76
Louvain	0.888	81.16
Infohiermap	0.846	75.95

The high performances reached by the classifiers built using the real values at time $\tau + 1$ indicate that a good approximation of forecasting methods to these values is fundamental to build reliable classifiers

Features forecast deviation We estimated how good is the proposed approach by analyzing the deviation of the values calculated with the forecasting methods with the real values of the features at $\tau + 1$. The models built using the real features at $\tau + 1$ reach good performances (see Table 12).

This indicates that a good approximation of the real values is important to build a reliable classifier. As a consequence of these good performances, an analysis of the deviation of the expected values obtained with time series forecast with the real values is needed to understand which measures can be predicted better than others with a certain community discovery algorithm or a certain forecasting technique. Thus, we analyzed the deviations $(f^{u,v}_{\tau+1} - \hat{f}^{u,v}_{\tau+1})^2$ of the expected values of the different forecasting methods with the real ones.

We analyzed the sum of squared error (SSE) for each forecasting method of each feature in Fig. 5, and we observed that: (1) DEMON and Infohiermap perform better with Ma, (2) Louvain is generally worse than the others for every feature, (3) Infohiermap works better for structural and topological (4), and DEMON minimizes the error for the community features. However, independently from the community discovery algorithm or the forecasting method, the deviation is always very low justifying the good performances previously exposed.

In particular, we found that, with respect to the other combinations, Infohiermap with LR has the highest SSE for each attribute. On the other hand, the best approximations are achieved by Infohiermap and DEMON with Ma with $n \in \{3, 4\}$. Indeed, with the exception of AA, Louvain never has the lowest SSE among the features used. At the same time, by ranking the SSE among the different community discovery algorithms and forecasting techniques, it emerges that with Louvain the lowest SSE belongs to AA while the highest to *SC*. On the contrary, with DEMON the lowest SSE belongs to *SC*, while the highest changes with respect to the forecasting method. Finally, as far as Infohiermap is concerned, we cannot derive nothing interesting. Thus, probably, due to its nature related to ego networks, DEMON gives better results than the other community discovery algorithms for community features, while AA works really well with the communities extracted by Louvain.

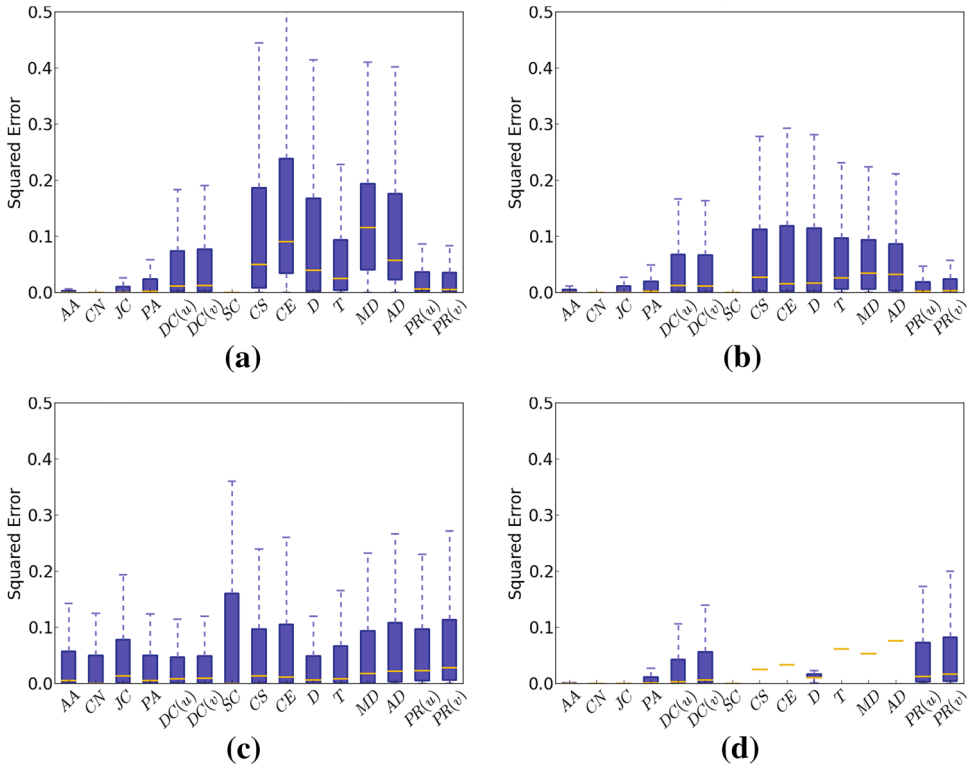

Fig. 5 Balanced scenario (Social). The *boxplots* of squared errors per feature show how independently from the community discovery algorithm or the forecasting method the deviation is always very low especially for the most important features **a** Social Louvain Ma, **b** Social Louvain LR, **c** Social DEMON Ma, **d** Social Infohiermap Ma

4.2.2 Unbalanced scenario

We have shown how the described analytic workflow is able to obtain good results when dealing with datasets having a balanced class distribution. Unfortunately, this scenario is not common when addressing the Interaction Prediction problem. Furthermore, making predictions on new interactions that will appear in a network involves, potentially, computing scores for all the $|V| \times (|V| - 1)$ pair of nodes of a network. Social networks are generally sparse, and this led to a high rate of false-positive predictions (in case of unsupervised approaches) or to models that maintain high accuracy just predicting the absence of new links (the majority classifier in case of supervised learning). Indeed, predicting every object as belonging to the most frequent class guarantee high performances, but in general it leads to useless classification results. For this reason, evaluating the performances of classifiers in highly unbalanced scenarios is not an easy task, but is definitely a very important one.

Since we want to predict correctly new links, our primary purpose is to reach high precision avoiding the generation of false-positive predictions. This is the reason why in the unbalanced scenario we will discuss, besides AUC and ACC, the *Lift Chart* and *precision* of the tested classifiers.

Precision is defined as $PPV = \frac{TP}{TP+FP}$. It represents the ratio of correct predictions for a specific class (in our case the one representing the presence of the edge in the test set) with respect to the total predictions provided.

Lift Chart graphically represents the improvement that a mining model provides when compared against a random guess, and measures the change in terms of lift score. By comparing the lift scores for various portions of a dataset and for different models, it is possible to determine which model is the best and which percentage of the cases in the dataset would benefit from applying the model's predictions.

We report the precision instead of the accuracy because, unlike the balanced scenario (where starting from a ratio of 50–50 the accuracy has a strong significance), in the unbalanced one it is very easy to get a high, but meaningless, accuracy. This is due to the fact that, as a consequence to the sparsity of the interaction network, the majority classifier can predict always "no edge" with no effort reaching very high performances. Besides this we report the *Lift Chart* because, conversely from *AUC* and *PPV* (with which shares, describing isomorphic spaces, the conveyed information), it is able, even in unbalanced scenarios, to graphically emphasize the improvements provided by the tested classifier against a baseline model.

We preserved the original ratio between the node pairs with and without a future interaction in Social and DBLP datasets. For both networks, we used the DEMON algorithm to extract communities. This choice is due to the following reasons: (1) *Social* DEMON reaches the best performances in the balanced scenario; thus, we expect that it will behave well even in unbalanced scenario; (2) *DBLP* using Louvain (i.e., the best performer in the balanced scenario) in the unbalanced scenario, all the classification models output the majority classifier.

In Social, the ratio of negative class to the total amount of possible pairs is 95.947 %, that means that a majority classifiers predicting no edge for all the pairs would have an accuracy of almost 96 %. As output from the classification phase with Ma, we have a model which reaches an AUC of 0.966 with a prediction accuracy of 98.75 % and a precision w.r.t. the positive class of 95.61 %. These two are very significant results: on the one hand, we have an accuracy improvement of 2.803 % in an ideal window of 4.053 % (100–95.947 %) with respect to the majority classifier while, on the other, we have a very high precision on the positive class, considering that a classifier predicting always an edge would have a precision of 4.053 %. In addition to the Ma model, we also built three classifiers each one of them considering all the forecasts for a single category of features: topological, structural and community.

In Fig. 6-left, we show the Lift Chart of the four models for Social. From the chart emerges that after the Ma model,

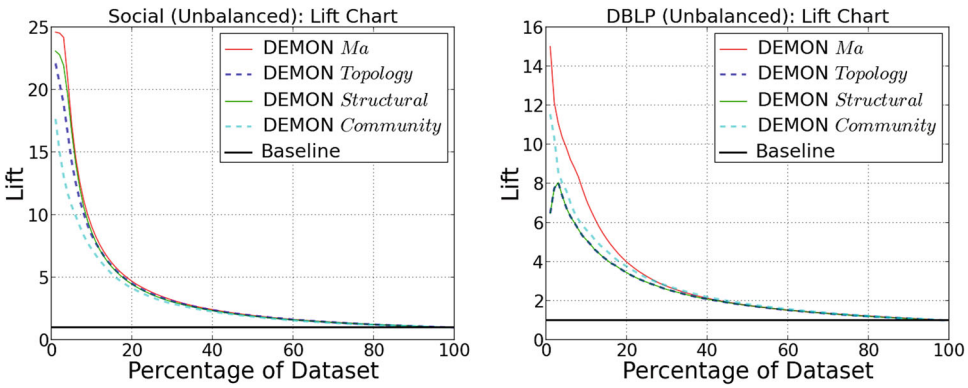

Fig. 6 Unbalanced scenario. The lift charts of the compared methods show how in both networks DEMON with Moving Average is the combination able to reach the best performances

Table 13 Unbalanced scenario (Social)

Algorithm	AUC	PPV (%)
SF Ma	0.897	64.06
SF LR	0.893	62.62
FSF Ma	0.918	**74.71**
FSF LR	**0.932**	72.45

Baselines on structural features using only Structural Forecast (*SF*) features calculated in the whole network and Filtered Structural Forecast (*FSF*) calculated following the proposed approach

In bold the AUC and PPV of the best performing approaches

the most promising is the one built upon the topological features followed by structural and community ones.

Also in this unbalanced scenario, we want to "measure" how much the community approach provides efficiency just filtering in the "promising pairs." By building the dataset with all possible pairs without leveraging community information, we get a majority class, i.e., the absence of a link, with a ratio of 98.96 % over the total number of entries. In order to better compare the two cases, we filter out randomly some pairs with no edge, bringing the accuracy of the majority classifier at 95.947 % (like in the case with community discovery). Again we compare the performances for the *SF* and *FSF*, which are reported in Table 13, but now considering the precision instead of the accuracy. We can see that we gain almost a 10 % of precision just filtering out, in any time slot, all the pairs not belonging to the same community. These results are very significant. On the one hand, we have an accuracy improvement of 2.803 % in an ideal window of 4.053 % (100–95.947 %) with respect to the majority classifier. On the other hand, we have a very high precision on the positive class, considering that a classifier predicting always an edge would have a precision of 4.053 %. In addition to the Ma model, we try to build also classifiers considering all the forecast methods but grouped for "kind of measure": topological, structural and community. It emerges that after the Ma model, the most promising is the one built upon the topological features followed by structural and community ones.

In DBLP case study, the resulting classifier has an AUC of 0.86, an ACC of 98.135 % and a precision with respect to the positive class of 44.78 %. The majority class (no link) has a ratio of 98.13 % over all the instances of the dataset. A possible reason for the lower performances obtained on DBLP w.r.t. Social is that in the latter an interaction represents a real social action between two different actors, while in DBLP an interaction models a relation of co-authorship in a paper, and the co-authorship is not, in our opinion, a strong representative of social interaction. However, we can notice that the performances are not completely bad: we have a precision of 44.78 %, starting from a ratio of positive class of 1.865 % (100–98.135 %), that is 24 times better than predicting for any pair the presence of the edge. Finally, we can observe from the Lift Chart in Fig. 6-right how, differently from the Social case, the most predictive set of features are the community ones, over the structural and topological.

4.3 Inter-community interaction prediction

So far, we have focused our attention on the task of predicting interaction within a community. We have shown that our approach is able to achieve good performances in case of both balanced and unbalanced class distributions and discussed the features that better predict the presence (or absence) of a new interaction. Here we address the complementary problem: prediction of *inter-community* interactions. Since the direct prediction of the network weak ties is a very complex problem prevalently due to the low stability of such links through time, we shift our interest to a related problem. We do not want to predict the specific endpoint of the interaction (i.e., user u of community C_j and user v of community C_z), but the presence of at least one interaction among users of two different communities, say C_j and C_z. To do so, we slightly modified our method:

- instead of using the original interaction network, we preprocess our data and build, for each snapshot, an induced graph using the previously extracted communities. In particular, for each snapshot graph G_i and related set of communities C_i we perform the transformation described by Algorithm 1;
- we compute the structural and topological features on the community-node pairs of each new induced graph;
- we apply the time series forecast and, on the forecasted feature values, we build the prediction model.

The main difference w.r.t. the original approach lies in the use of the communities as network nodes and not as filters (i.e., no community features are used to build the final model).

Algorithm 1 BuildInducedGraph(G_i, C_i)

Require: G_i: network snapshot, C_i: community set.
1: CoreNodes = {}
2: IG = NEWGRAPH
3: **for** $c \in C_i$ **do**
4: c_{cores} = IDENTIFYCOMMUNITYCORES(C_i)
5: CoreNodes[c] = c_{cores}
6: **end for**
7: **for** $c_j \in C_i$ **do**
8: **for** $c_z \in C_i$ **do**
9: **if** $c_j \neq c_z$ **then**
10: **if** $\exists(u, v) \in G_i, u \in CoreNodes[c_j], v \in CoreNodes[c_z]$ **then**
11: IG.ADDEDGE(c_j, c_z)
12: **end if**
13: **end if**
14: **end for**
15: **end for**
16: **return** IG

A crucial aspect is the process used to build each community-graph. As shown in Algorithm 1, for each community are identified the core nodes (lines 3-6): then, a new edge is created in the induced graph among the community C_j and C_z if there exists at least one edge in the original graph connecting two of their core nodes (lines 7-15). There are several ways to implement the IDENTIFYCOMMUNITYCORES function: in our experiments, we use the top-k% high-degree nodes within each community (we fixed k to 5). After the construction of the community-network, we apply a reconciliation phase across consecutive snapshots in order to align the community ids. To build the evolutive chain of each community (i.e., to find the correspondence of a given community across time), we employed a well-established set matching procedure often used by dynamic community discovery approaches (Hartmann et al. 2014), namely the Jaccard matching:

$$\text{Jaccard}(C_t, C_{t+1}) = \frac{|\bigcap(C_t, C_{t+1})|}{|\bigcup(C_t, C_{t+1})|} \quad (1)$$

Given a community C at time t (C_t in the equation), we identify as its future expression in $t+1$ the community which maximizes the Jaccard function upon their node sets. We decided to evaluate the introduced methodology on a very specific case study: inter-community interaction prediction on the DBLP community-graph built upon the Infohiermap partition. The reasons behind such choice are the following:

- Among the previously analyzed datasets, DBLP is the bigger one and it is always decomposed in a higher number of communities (ensuring community-graphs of meaningful size);
- DEMON generates overlapping communities; thus, the community-graph extraction loses some effectiveness (shared nodes generate a densely connected graph);
- Louvain as all modularity-based approaches suffers from the scale problem: this causes very sparse star-like community-graphs composed by few focal nodes (i.e., the bigger communities) linked to many satellites (i.e., very small communities that are rarely connected by interactions).

4.3.1 Balanced scenario

In the *intra-community* scenario, w.r.t. the DBLP dataset and Infohiermap communities, we were able to produce predictions for, approximately, the 91 % of the interactions actually present in the test set. The filter produced by the application of Infohiermap was then able to discriminate weak ties across different network partitions and guarantee high AUC and Accuracy. Due to the community-graph construction defined in Algorithm 1 we now group together the remaining 9 % of the interactions in meta-links connecting different Infohiermap communities. Obviously, due to the IDENTIFYCOMMUNITYCORES strategy, we will not able to make prediction for all the weak ties: however, the filtering introduced groups together 97 % of them producing a very reliable sample.

Following the method designed for *inter-community* interaction prediction, we tested all the different time series forecasting strategies discussed in Table 4 and defined as Ma the one having high score (as shown in Fig. 7 for both the balanced and unbalanced scenarios). On the balanced class scenario, we obtained the results reported in Table 14. Our results are, as expected, not as good as the one obtained for the *intra-community* interaction problem. Here the best predictive power is expressed by the Ma time series forecast able to reach 66 % of accuracy w.r.t. the 50 % of the majority classifier. In order to better understand the impact of the time variable on such very volatile network structure, we also trained a classifier on the same

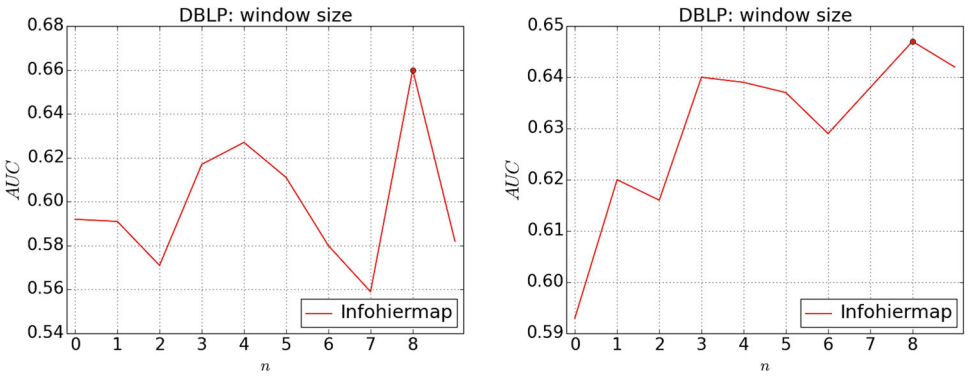

Fig. 7 Inter-community prediction: *left* balanced and *right* unbalanced scenarios. *AUC* values varying $n \in [0, \tau]$ using the Moving Average Ma. Dots highlight highest values. In both scenarios, the optimal window size is 8

Table 14 Balance scenario (DBLP)

Algorithm	AUC	ACC (%)
Lv	0.580	56.01
Avg	0.650	65.10
Ma	**0.660**	66.00
LR	0.581	58.10
Flat Graph	0.610	59.12
Baseline	0.500	50.00

Infohiermap performances for the inter-community prediction. The Moving Average Ma forecasted features allow for the best classification models

In bold the AUC of the best performing approach

Table 15 Unbalanced scenario (DBLP)

Algorithm	AUC	PPV (%)
Lv	0.594	33.33
Avg	0.632	07.02
Ma	**0.647**	50.00
LR	0.596	50.00
Flat Graph	0.316	57.20
Baseline	0.504	4.01

Infohiermap performances for the inter-community prediction. Like in the balanced scenario, the Moving Average Ma forecasted features allow for the best classification models

In bold the AUC of the best performing approach

feature set computed on the flattened community-graph (i.e., the graph built by keeping together nodes and edges of all the temporal snapshots). The obtained results suggest us that conversely from the *intra-community* settings here the adoption of time series does not play a crucial role even though it allows with Ma and Avg forecasting to slightly increase the prediction accuracy.

4.3.2 Unbalanced scenario

To complete our analysis, we evaluated the effectiveness of our approach even in the unbalanced *inter-communities* setting. This scenario represents the most complex one we can design: we are targeting weak ties (i.e., the 9 % of the interactions not covered by the *intra-community* predictions) when the majority class—no interaction—is approximately 98 %.

The results in Table 15 show a relatively high precision w.r.t. the minority class: while the baseline (the minority classifier) reaches 4.01 % precision, our approach is able to reach $PPV = 50\,\%$ (even though the recall on the minority class drops from 100 % to "only" 65 %). Even in this scenario, the Ma time series forecast strategy is the one that offers higher quality models. Conversely from the balanced scenario, we can observe how the classifier built upon the flattened community-graph does not produce interesting results: even though it guarantees higher precision ($PPV = 57.2\,\%$) the overall model quality is lower (Flat graph $AUC = .316$ vs. Ma $AUC = .647$). The predictions made on the flattened networks are more precise, but the recall is low ($\sim 9\,\%$). In an unbalanced scenario, the low stability of *inter-community* interactions amplifies the complexity of the predictive task: flattening the temporal dimension causes an increase of the false-negative predictions, which leads to performance degradation.

5 Related works

In the literature, there is a wide study of the link prediction problem. The methods used to solve LP apply supervised and/or unsupervised approaches (Lü and Zhou 2011). In particular, link prediction strategies may be broadly categorized into four groups: *(q)* similarity-based strategies, (2) maximum likelihood algorithms, (3) probabilistic models and (4) supervised learning algorithms (Lü and Zhou 2011).

The first group defines measures of similarity as a score between each pair of nodes. All non-observed links are ranked according to their scores, and the links connecting more similar nodes are supposed to be of higher existence likelihoods. Despite its simplicity, the definition of node similarity is a non-trivial challenge. A similarity index can be very simple or very complicated, and it may work well for some networks while fail for some others. For example, in Dong et al. (2012) the authors introduce a unsupervised method based on ranking factors using the assumption that people make friends in different networks following similar principles.

The second set of methods is based on maximum likelihood estimation. Empirical studies suggest that many real-world networks exhibit hierarchical organization. These algorithms presuppose some organizing principles of the network structure, with the detailed rules and specific parameters obtained by maximizing the likelihood of the observed structure. From the viewpoint of practical applications, an obvious drawback of the maximum likelihood methods is that it is very time-consuming. In addition, the maximum likelihood methods are probably not among the most accurate ones. Huang et al. (2012) use continuous-time stochastic process for predicting aggregate social activities, that is different activities between users in the same social network.

The third group of algorithms is based on probabilistic Bayesian estimation. Probabilistic models aim at abstracting the underlying structure from the observed network, and then predicting the missing links by using the learned model. Given a target network, the probabilistic model will optimize a built target function to establish a model based on a group of parameters, which can best fit the observed data of the target network. Then the probability that a nonexistent link will appear is estimated by the conditional probability. In Zhu (2012) is proposed a way to develop nonparametric latent feature relational models to minimize an objective function for a normalized link likelihood model.

The proposed approach belongs to the category of methods which employ supervised machine learning techniques. LP through supervised learning algorithms was introduced in Liben-Nowell and Kleinberg (2007). The authors studied the usefulness of graph topological features by testing them on co-authorship networks. A classifier is trained according to the knowledge that a link will be present or not in future. Then the classifier is used to predict new links. After Liben-Nowell and Kleinberg (2007), a wide range of models exploiting several different strategies have been proposed. Indeed, there has been proved that supervised methods reach better performances than unsupervised ones, in terms of both AUC and precision.

In order to build an efficient classifier, many works focused on finding an efficient set of features. In Jahanbakhsh et al. (2012) is shown that only a small set of features are essential for predicting new edges and that contacts between nodes with high centrality are more predictable than nodes with low centrality. Following these principles, in Bao et al. (2013) principal component analysis is used to determine the weights of the features. According to these weights is reduced the number of features taken in input by the regression algorithm used for prediction. A rank aggregation approach is proposed in Pujari and Kanawati (2012). The authors rank the list of unlinked nodes according to some topological measures, then at the new instant time each measure is weighted according to its performance in predicting new links. The learned weights are used in a reinforcing way for the final prediction. Finally, in Spiegel et al. (2011) tensor factorization is used to select the more predictive attributes, while in Lichtenwalter et al. (2010) important features for link prediction are examined and it is provided a general, high-performance framework for the prediction task.

Like we did with community features, many works reinforce the classifier with other kind of knowledge. The authors of Shibata et al. (2012) used textual features besides the topological ones and applied SVM as supervised learning method. In Wang et al. (2011), spatial and mobility information are used to help the classifier.

Despite the good performances achieved, all the works reported until now do not solve the interaction prediction problem. Some works which consider dynamic networks are Bringmann et al. (2010) and Bliss et al. (2013). In Bringmann et al. (2010), association rules and frequent-pattern mining are used to search for typical patterns of structural changes in dynamic networks. The authors developed the Graph Evolution Rule Miner to extract such rules and applied these rules to predict future network evolution. In Bliss et al. (2013), the prediction is optimized through weights which are used in a linear combination of sixteen neighborhoods and node similarity features by applying the covariance matrix adaptation evolution strategy. However, in this second work the authors tried to predict only new interactions and not re-occurring ones. Finally, other works like da Silva Soares and Prudencio (2012), Sarkar et al. (2012) show how an approach based on time series modeling the evolution of continue univariate features describing node characteristics substantially helps in solving the link prediction task.

As shown in Lü and Zhou (2011), despite the high precision, supervised approaches can be prohibitively time-consuming for a large networks having over 10, 000 nodes. Moreover, supervised methods are proved to reach better performances in terms of both accuracy and precision than unsupervised methods. Thus, given our interest in large, sparse networks, and given that all the works cited highlight the importance of using features outside the links'

dimension, our focus on local information gathered from communities and time series features to train the classifier is justified. In order to reduce the computational complexity, several approaches such as Soundarajan and Hopcroft (2012) make use of clustering and community information. These analyses suggest that clustering information, no matter the algorithm used, improves link prediction accuracy.

In order to build an efficient classifier for link prediction, it is crucial to define and calculate a set of graph structural features. As stated by the papers mentioned previously, when dealing with large-scale graphs that may include millions of vertexes and links, one of the challenges is the computationally intensive extraction of such features. Using our approach, we dramatically reduce the features computation because the calculus is performed considering separately the links present in network's communities. Several studies related to link prediction such as Feng et al. (2012), Fire et al. (2013), Jahanbakhsh et al. (2012), Lichtnwalter and Chawla (2012), Xu and Rockmore (2012) try to suggest which are optimal topological structures of a network and the best features to be used with. For example, in Feng et al. (2012) it is analyzed the relation between network structure and the performance of link prediction algorithm, while in Jahanbakhsh et al. (2012) it is shown that only a small set of features are essential for predicting new edges and that contacts between nodes with high centrality are more predictable than nodes with low centrality. The authors finally claim that on networks with low clustering coefficient, link prediction methods perform poorly, while, as the clustering coefficient grows, the accuracy is drastically improved. Fire et al. (2013) investigate the effectiveness of link prediction by gradually reducing the number of visible links in the studied networks. They demonstrate that classification quality degrades with the number of visible links and that a small fraction of visible links helps in solving the problem with chances significantly higher than random. The authors of Xu and Rockmore (2012) propose a feature selection framework based on ranking, weighting, correlation and redundancy. In particular, they focus on preserving the maximum accuracy by finding the minimum redundancy in the feature space by using a greedy scheme.

We proved that a specific community discovery algorithm can improve the performances depending on the type of dataset. Moreover, the main difference between our approach and those of the works reported is that our prediction is based not only on the observed structural, topological and community features, but also on the forecast of the future features. In other words, it improves the state of the art by combining the use of community and time series for solving interaction prediction.

Finally, in the literature there are only few works treating the problem of weak ties in link prediction that we analyzed in the last section. Some studies show how and why weak ties can be useful in link prediction. In particular in Lü and Zhou (2009) is shown how the accuracy in link prediction can be improved by exploiting the contribution of weak ties. The *Weak Ties Theory* (Granovetter 1973) states that people usually obtain useful information or opportunities through the acquaintances often not the close friends, i.e., the weak links in their friendship network play a significant role. Recently, the authors of Onnela et al. (2007) demonstrated that the weak ties mainly maintain the connectivity in mobile communication networks, and in Csermely (2004) is explained how weak ties maintain the stability of biological systems. In Xiang et al. (2010) is developed an unsupervised model to estimate relationship strength from interaction activity and user similarity, while in Gilbert and Karahalios (2009) is presented a predictive model that maps social media data to tie strength. These approaches were not exploited nor used in our workflow on weak ties because of (1) the dynamic nature of our dataset, (2) the higher abstraction level selected (i.e., we consider weak ties as the ties among communities and we loose the original source and destination node), (3) we wanted to replicated the workflow adopted for link prediction of strong ties.

6 Conclusions

In this work, we have tackled the Link Prediction problem in a dynamic network scenario. Since networks often model highly evolving realities that cannot easily be *"frozen"* in time without loss of information, a time-aware approach to link prediction is mandatory to achieve valuable results. Moreover, due to the intrinsic high computational cost of the approaches that solve this problem, it is important to reduce the list of possible candidates for which to compute a prediction (preferably avoiding the generation of false positives). To this extent, we have exploited the community structure of social networks to both bound the result set and design features whose analysis through time have allowed the description of a high-performance supervised learning strategy. Anyhow, using network partitions as filters make the proposed approach focus only on the prediction of *intra-community* interactions: to overcome this issue, we propose an experimental setting specifically designed to address *inter-community* interaction prediction. Using community-induced graphs, we show that the proposed analytical workflow can be applied to this complex problem and discuss the quality of the obtained results.

The results obtained with the proposed methodology open the way to several future lines of analysis. Indeed, more accurate time series forecast techniques can be evaluated in order to reduce the forecast error and evolutionary community discovery approaches can be used in order to incorporate communities life cycle features within the predictive process. Moreover, with respect to the type of dataset used, it could be possible to consider other types of features such as mobility knowledge and spatial co-location. All these improvements will lead to more narrow and sophisticated classifiers that, taking into account more and more aspects, will be able to better predict future human interactions.

Acknowledgments This work was partially funded by the European Community's H2020 Program under the funding scheme "FETPROACT-1-2014: Global Systems Science (GSS)," Grant agreement # 641191 CIMPLEX4 "Bringing CItizens, Models and Data together in Participatory, Interactive SociaL EXploratories," https://www.cimplex-project.eu. This work is supported by the European Community's H2020 Program under the scheme "INFRAIA-1-2014-2015: Research Infrastructures," Grant agreement #654024 *"SoBigData: Social Mining & Big Data Ecosystem,"* http://www.sobigdata.eu.

References

Adamic LA, Adar E (2003) Friends and neighbors on the web. Soc Netw 25(3):211–230

Bao Z, Zeng Y, Tay YC (2013) sonLP: social network link prediction by principal component regression. In: IEEE/ACM international conference advances in social networks analysis and mining (ASONAM). IEEE, pp 364–371

Barabási A-L, Albert R (1999) Emergence of scaling in random networks. Science 286(5439):509–512

Bliss CA, Frank MR, Danforth CM, Dodds PS (2013) An evolutionary algorithm approach to link prediction in dynamic social networks. arXiv:1304.6257

Blondel VD, Guillaume J-L, Lambiotte R, Lefebvre E (2008) Fast unfolding of communities in large networks. J Stat Mech Theory Exp 2008(10):P10008

Bringmann B, Berlingerio M, Bonchi F, Gionis A (2010) Learning and predicting the evolution of social networks. IEEE Intell Syst 25(4):26–35

Coscia M, Rossetti G, Giannotti F, Pedreschi D (2012) Demon: a local-first discovery method for overlapping communities. In: Proceedings of the 18th ACM SIGKDD international conference on knowledge discovery and data mining. ACM, pp 615–623

Csermely P (2004) Strong links are important, but weak links stabilize them. Trends Biochem Sci 29(7):331–334

da Silva Soares PR, Prudencio RBC (2012) Time series based link prediction. In: IEEE international joint conference on neural networks (IJCNN). doi:10.1109/IJCNN.2012.6252471

Dong Y, Tang J, Wu S, Tian J, Chawla NV, Rao J, Cao H (2012) Link prediction and recommendation across heterogeneous social networks. In: 2012 IEEE 12th international conference on data mining (ICDM). IEEE, pp 181–190

Feng X, Zhao J, Xu K (2012) Link prediction in complex networks: a clustering perspective. Eur Phys J B 85(1):1–9

Fire M, Puzis R, Elovici Y (2013) Link prediction in highly fractional data sets. In: Subrahmanian VS (ed) Handbook of computational approaches to counterterrorism. Springer, New York, pp 283–300

Gilbert E, Karahalios K (2009) Predicting tie strength with social media. In: Proceedings of the SIGCHI conference on human factors in computing systems. ACM, pp 211–220

Girvan M, Newman ME (2002) Community structure in social and biological networks. Proc Natl Acad Sci 99(12):7821–7826

Granovetter M (1973) The strength of weak ties. Am J Sociol 78(6):1

Hartmann T, Kappes A, Wagner D (2014) Clustering evolving networks. arXiv:1401.3516

Huang S, Chen M, Luo B, Lee D (2012) Predicting aggregate social activities using continuous-time stochastic process. In: Proceedings of the 21st ACM international conference on information and knowledge management. ACM, pp 982–991

Jahanbakhsh K, King V, Shoja GC (2012) Predicting human contacts in mobile social networks using supervised learning. In: SIMPLEX workshop, ACM

Liben-Nowell D, Kleinberg J (2007) The link prediction problem for social networks. J Am Soc Inform Sci Technol 58(7):1019–1031

Lichtenwalter RN, Lussier JT, Chawla NV (2010) New perspectives and methods in link prediction. In: Proceedings of the 16th ACM SIGKDD international conference on knowledge discovery and data mining. ACM, New York, pp 243–252. doi:10.1145/1835804.1835837

Lichtnwalter R, Chawla NV (2012) Link prediction: fair and effective evaluation. In: 2012 IEEE/ACM international conference on advances in social networks analysis and mining (ASONAM). IEEE, pp 376–383

Lü L, Zhou T (2009) Role of weak ties in link prediction of complex networks. In: Proceedings of the 1st ACM international workshop on complex networks meet information & knowledge management. ACM, pp 55–58

Lü L, Zhou T (2011) Link prediction in complex networks: a survey. Phys A Stat Mech Appl 390(6):1150–1170

Newman MEJ (2001) Clustering and preferential attachment in growing networks. Phys Rev E 64(2):025102

Onnela J-P, Saramäki J, Hyvönen J, Szabó G, Lazer D, Kaski K, Kertész J, Barabási A-L (2007) Structure and tie strengths in mobile communication networks. Proc Natl Acad Sci 104(18):7332–7336

Page L, Brin S, Motwani R, Winograd T (1999) The pagerank citation ranking: bringing order to the web. Technical Report 1999-66, Stanford InfoLab, November 1999

Pujari M, Kanawati R (2012) Supervised rank aggregation approach for link prediction in complex networks. In: Proceedings of the 21st ACM international conference on World Wide Web, pp 1189–1196

Rapoport A (1963) Mathematical models of social interaction. In: Luce et al (eds) Handbook of mathematical psychology, vol 2. Wiley, New York

Rosvall M, Bergstrom CT (2011) Multilevel compression of random walks on networks reveals hierarchical organization in large integrated systems. PloS one 6(4):e18209

Salton G, McGill MJ (1983) Introduction to modern information retrieval. McGraw-Hill, New York

Sarkar P, Chakrabarti D, Jordan M (2012) Nonparametric link prediction in dynamic networks. arXiv:1206.6394

Shibata N, Yuya K, Ichiro S (2012) Link prediction in citation networks. J Am Soc Inform Sci Technol 63(1):78–85

Soundarajan S, Hopcroft J (2012) Using community information to improve the precision of link prediction methods. In: Proceedings of the 21st ACM international conference on World Wide Web, pp 607–608

Spiegel S, Clausen J, Albayrak S, Kunegis J (2011) Link prediction on evolving data using tensor factorization. In: Pacific-Asia conference on knowledge discovery and data mining. Springer, Berlin, Heidelberg, pp 100–110

Wang D, Pedreschi D, Song C, Giannotti F, Barabasi AL (2011) Human mobility, social ties, and link prediction. In: Proceedings of the 17th ACM SIGKDD international conference on knowledge discovery and data mining, pp 1100–1108

Watts DJ, Strogatz SH (1998) Collective dynamics of 'small-world'networks. Nature 393(6684):440–442

Xiang R, Neville J, Rogati M (2010) Modeling relationship strength in online social networks. In: Proceedings of the 19th international conference on world wide web. ACM, pp 981–990

Xu Y, Rockmore D (2012) Feature selection for link prediction. In: CIKM workshop. ACM

Zhu J (2012) Max-margin nonparametric latent feature models for link prediction. arXiv:1206.4659