Combining structural and dynamic information to predict activity in link streams

Thibaud Arnoux, Lionel Tabourier, Matthieu Latapy

Combining structural and dynamic information to predict activity in link streams

Thibaud Arnoux, Lionel Tabourier, Matthieu Latapy.
Sorbonne Universités, UPMC Univ Paris 06, CNRS, LIP6 UMR 7606,
4 place Jussieu 75005 Paris
http://www.complexnetworks.fr
Email: firstname.lastname@lip6.fr

June 29, 2017

Abstract

A link stream is a sequence of triplets (t, u, v) meaning that nodes u and v have interacted at time t. Capturing both the structural and temporal aspects of interactions is crucial for many real world datasets like contact between individuals. We tackle the issue of activity prediction in link streams, that is to say predicting the number of links occurring during a given period of time and we present a protocol that takes advantage of the temporal and structural information contained in the link stream. We introduce a way to represent the information captured using different features and combine them in a prediction function which is used to evaluate the future activity of links.

1 Introduction

A link stream (see Figure 1) is a sequence of triplets (t, u, v), each triplet indicating that an interaction occurred between u and v at time t. Many real world datasets can be modeled and analyzed using link streams, such as e-mail exchanges, contacts between individuals, phone calls or IP traffic [1]. There have been other attempts to model these systems, like dynamical networks [2] or time varying graphs [3], which hold the same information as link streams. Analyzing the dynamical and structural properties of these link streams is capital to apprehend the behavior of the system, as it allows to understand the underlying phenomena in the data.

We focus on the activity prediction problem, *i.e.* predicting the number of links appearing between each pair of nodes during a given period of time. While this problem shares properties with the more usual link prediction problem, it is also quite different in the sense that we aim at predicting not only who interacts with who, but also when.

To do so, we capture independently some structural and dynamical features with metrics measuring the link stream properties. We then combine these metrics in order to estimate future activity and we compare our prediction to the ground truth in order to assess the relevance of the approach. The performance of our framework is measured on two datasets of real world contacts between individuals [4, 5].

Let us emphasize the fact that our goal is to define a general framework to allow further study of the interplay between structural and dynamical features for prediction tasks, rather than optimize the prediction performance on these specific datasets. We aim to give evidence of the fact that combining these two kinds of features leads to improvements over the use of only one kind. Therefore, we make the simplest possible design choices in order to make our point, even though more elaborate choices would lead to better predictions. Our work is indeed a proof of concept, that provides a general scheme to serve as a baseline and motivation for further work in this direction.

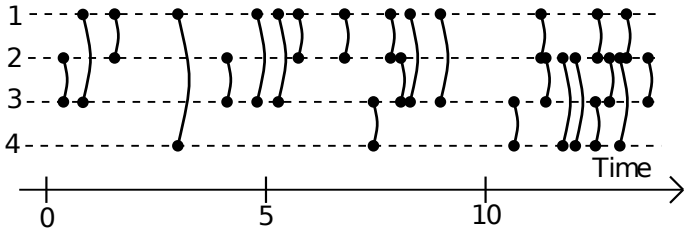

Figure 1: Example of a link stream where nodes 2 and 3 have interacted at time 0.5, nodes 1 and 3 have interacted at time 1, and so on.

2 Related work

Activity prediction is related to the classic link prediction problem, which consists in using the data structure by representing it as a graph [6, 7], and predict new links appearing in this graph. When the temporal evolution is a key element in the data, a usual approach is to slice the data in several time windows T_i, then aggregate them as a sequence of graphs $G_i = (V, E_i)$ corresponding to the time windows such that $E_i = \{(u, v) : \exists (t, u, v) \in E\}$. It allows to use traditional link prediction methods on these graphs. The information contained in the data is then extracted using graph-based measurements. In this field, many metrics have been developed to obtain the most relevant information [8, 9]. They often consist in evaluating the similarity between two nodes according to various criteria, which produces a score or a ranking correlated to the apparition probability of a link between these nodes. For example, the number of common neighbors between two

nodes and several variants [10] are commonly used. Similarity measures based on the temporal patterns of activities of nodes and links have also been proposed (*e.g.* [11]). Several methods exist to combine the metrics computed for improving the prediction. It is possible to use classification algorithms to determine the predicted links [8]. Another approach is to rank the pairs of nodes using the values of different metrics. The predicted links connect the n first pairs of nodes, with n fixed as a parameter and determined using the system behavior [12, 13]. However, the use of time windows commands a time scale and leads to the loss of some temporal information. For example, the information associated to a link repetition between two nodes within a time window disappear during the data aggregation. One of the stakes of our work is to conserve this information by using the link stream formalism, more suited to the data.

It is also possible to approach link prediction by focusing on the dynamical aspects of the link apparition between two nodes rather than on the structural properties. The sequence of links between each pair of nodes is then considered as a time series and numerous tools have been developed in this field to predict the future behavior of the system. For example it is possible to focus on the link apparition frequency in the past to predict future interactions [14, 15]. This approach focuses on predicting the future occurrences of links that have appeared in the past. As such, it is complementary to link prediction in graphs.

Our work differs from these methods by focusing on both the dynamic and the structural aspect of the data while avoiding the information loss induced by the use of time windows. We introduce a protocol that combines these information sources and allows a fine temporal resolution. We use this to predict both new and repeated links in the stream.

3 Problem definition

We consider a set of nodes V representing entities in the system – e.g., individuals in contact networks or mobile devices in DTN. We observe interactions between these entities for a period of time $T = [A, \Omega]$, that we model as a link stream $L = (T, V, E)$, where $E \subseteq T \times V \times V$, and $(t, u, v) \in E$ means that an interaction occurred between u and v at time t. In the following, we refer to L and T as the *observation stream* and *observation period*, respectively. Our goal is to predict the number of interactions between nodes in V during another period of time $T' = [A', \Omega']$ with $\Omega \leq A' < \Omega'$. We model the interactions during this interval as a link stream $L' = (T', V, E')$ with $E' \subseteq T' \times V \times V$. L' and T' are then called the *prediction stream* and *prediction period*. We aim to predict the activity of any pair of nodes in the system, *i.e.* for each $(u, v) \in V \times V$, the value $\mathcal{A}'(u, v) = |\{(t, u, v) \in E'\}|$.

Table 1: Graph-based metrics used for prediction.

Common Neighbors	$\lvert\mathcal{N}(u) \cap \mathcal{N}(v)\rvert$	Proximity of two nodes based on the number of common neighbors
Jaccard	$\frac{\lvert\mathcal{N}(u)\cap\mathcal{N}(v)\rvert}{\lvert\mathcal{N}(u)\cup\mathcal{N}(v)\rvert}$	Decrease the weight of high degree nodes
Adamic Adar [16]	$\sum_{w\in\mathcal{N}(u)\cap\mathcal{N}(v)} \frac{1}{\log\lvert\mathcal{N}(w)\rvert}$	Decrease the weight of nodes with high degree neighborhood

4 Activity prediction framework

4.1 Prediction features

The information contained in a link stream can be of different kinds, for instance, it can be the number of past interactions between two nodes or the density of a node's neighborhood. While existing methods mostly focus on one of these aspects, we intend at using metrics adapted to link streams, which combine dynamical and structural information. The metrics represent complex characteristics of the system, however expressing them with a scalar score leads to a loss of a part of the information. For example, a metric accounting for the burstiness of the interactions between two nodes is poorly represented by a unique score. Hence, we define functions that capture different aspects of the information. We want these functions to represent how likely is a link apparition between each pair of nodes according to each metric. We then combine them to form the prediction function itself.

4.1.1 Structural measurements

As a first step, we adapt metrics from link prediction in graphs to the context of activity prediction in link streams. We use traditional metrics widely employed in link prediction: the number of common neighbors between two nodes u and v, and other derived metrics, *i.e.* the Jaccard index or the Adamic-Adar index (see Table 1). Their definitions use the notion of neighborhood of a node u in L as $\mathcal{N}(u) = \{v : \exists(x, u, v) \in E\}$. As these metrics do not take into account the dynamics of the data, we represent them as a function independent of t. Thus, the function associated to the number of common neighbors of two nodes u, v is $\lvert\mathcal{N}(u)\cap\mathcal{N}(v)\rvert$. We define similarly the function associated with the Jaccard index and the Adamic-Adar index.

4.1.2 Temporal measurements

The link stream formalism allows to capture temporal information in the data, for this purpose we restrict ourselves to relatively simple measurements. As it is usually done in the field of time-series prediction, we first use as a benchmark a metric based on the extrapolation of the activity between each pair of nodes, which allows to take into account link repetitions in the stream. We want to keep the benchmark as simple as possible. Therefore, we choose to represent the benchmark with a time-independent function. Precisely, the number of interactions $\mathcal{A}_{u,v}$ between u and v during T, is defined the function $\mathcal{A}_{u,v}(t) = |\{(x, u, v) \in E\}|$.

Then, we define three other metrics that describe more precisely the temporal behaviors of the system. The first two are adapted versions of the pair activity extrapolation that focus on the most recent activity during the observation period. This choice is made on the ground that the most recent interactions affect more the dynamics than the old ones do.

In one case, we only take into account the activity during a fixed period of time: for each pair of nodes, we compute the function $\mathcal{A}_{\delta,(u,v)}(t) = |\{(x, u, v) \in E \ : \ x \in [\Omega - \delta, \Omega]\}|$, which accounts for the number of interactions during the most recent δ period.

In the other case, we take into account the activity of each pair of nodes between Ω and the time of occurrence of the k^{th} link between u and v before Ω. The corresponding function is $\mathcal{A}_{k,(u,v)}(t) = k/(\Omega - t_k)$ with t_k such that $|\{(x, u, v) \in L, \Omega \geq x \geq t_k\}| = k$. Note that both of these functions are time-independent in order to be easily comparable with the benchmark.

Finally, we define a temporal metric that aims at taking into account the variations of activity during the observation period. The observation period is divided equally in n sub-periods. For each pair of nodes we then fit an affine function using the activity over each sub-period. The corresponding function is the extrapolation of the affine function $\mathcal{A}_{n,(u,v)}(t) = a_{u,v} \cdot t + b_{u,v}$, with $a_{u,v}$ and $b_{u,v}$ the coefficient computed by the fitting algorithm for each pair u, v. This metric allows to study the behavior of time-dependent prediction functions.

4.2 Prediction function

To use the information captured by the metrics presented above, we choose a method to combine the functions. We construct a prediction function $f_{u,v}$, such that for all $u, v \in V \times V$, and for all $t \in [A', \Omega']$, $f_{u,v}(t)$ represents how likely is the apparition of a link between u and v at the time t. We build it from the metric functions using a linear combination:

$$f_{u,v}(t) = \sum_{i=1}^{k} \alpha_i \cdot f^i_{u,v}(t)$$

$f^i_{u,v}$ is the function associated to metric i and k is the number of metrics used. The parameters α_i allow to control each metric weight in the prediction function. Note that the value of $f_{u,v}$ does not have an absolute meaning, we rather use the relative value of $f_{u,v}$ to other $f_{u',v'}$ as we will see in Section 4.3. Note also that other combination methods are possible, this choice is made for the sake of simplicity.

Given such a prediction function, a standard prediction method consists in learning on a training period the α_i values which optimize a given evaluation criterion. Then these weights are used for the actual prediction. In the following, we explore the influence of the weights on cases where the sum has two terms, we do not focus on a specific learning method which is left for future works.

4.3 Prediction protocol

To predict the number of links between each pair of nodes during $T' = [A', \Omega']$, we first estimate the number of links N between all pairs of nodes during this period. We make the strong assumption that the global activity in L' is the same as in L, and therefore, extrapolate linearly the stream activity to determine the global number N of links to predict:

$$N = |E| \cdot \frac{\Omega' - A'}{\Omega - A} \tag{1}$$

Then, for each pair of nodes, the function f^i are computed on the link stream L. As the prediction function reflects how likely is the occurrence of a link between two nodes, we use it to allocate the N links between all the pairs in $V \times V$. We define the *pair apparition score* $\Gamma_{u,v}$ as a score of the link apparition likelihood between u and v during T':

$$\Gamma_{u,v} = \int_{A'}^{\Omega'} f_{u,v}(t)\, dt \tag{2}$$

It is then normalized to the sum of all $\Gamma_{x,y}$ for all pairs of nodes x, y in the stream. We allocate the N links estimated previously proportionally to the normalized pair apparition score to get $N_{u,v}$, the number of interactions predicted for any pair (u, v):

$$N_{u,v} = N \cdot \frac{\Gamma_{u,v}}{\sum_{x,y \in V} \Gamma_{x,y}} \tag{3}$$

As mentioned previously, it is the relative value of $\Gamma_{u,v}$ which allows to compute the $N_{u,v}$, with $\sum_{x,y \in V} N_{u,v} = N$.

This framework allows to predict the number of links between each pair of nodes during T'. It is important to note that due to the specificities of our prediction task and in contrast to what is usually done for link prediction in graphs, this number is not necessarily an integer.

4.4 Evaluation protocol

As our method aims to predict a different object from link prediction methods in graphs, we have to define another way to evaluate the efficiency of our protocol. Nevertheless, the evaluation method defined here aims to stay as close as possible to the tools used in classification tasks, therefore allowing to compare our method to other prediction algorithms. We adapt the usual definition of true positives, false positives and false negatives to the context of activity prediction in link streams. Precisely, for each pair (u, v), we compare $N_{u,v}$, the number of links predicted, to $N'_{u,v}$, the number of links that have actually occurred between A' and Ω' (see figure 2). We then define the number of TP, FP and FN as follows:

$$\begin{cases} TP_{u,v} = \min(N_{u,v}, N'_{u,v}) \\ FP_{u,v} = \max(N_{u,v} - N'_{u,v}, 0) \\ FN_{u,v} = \max(N'_{u,v} - N_{u,v}, 0) \end{cases}$$

The sum of each of these indicators over all pairs of nodes yields the number of TP, FP and FN for the whole prediction. Note that these definitions allow to get the usual relationships between the indicators, that is to say, $TP + FP$ is the number of predictions and $TP + FN$ is the total number of interactions actually occurring during T'.

Thus we can compute more sophisticated performance measurements: the precision $\left(\frac{TP}{TP+FP}\right)$, the recall $\left(\frac{TP}{TP+FN}\right)$. We also use the F-score to quantify the quality of prediction, which is the harmonic mean of these two indicators: $2 \cdot \frac{precision \cdot recall}{precision + recall}$. Other indicators could be defined in this context, like the ROC curve, but we do not use them in this study.

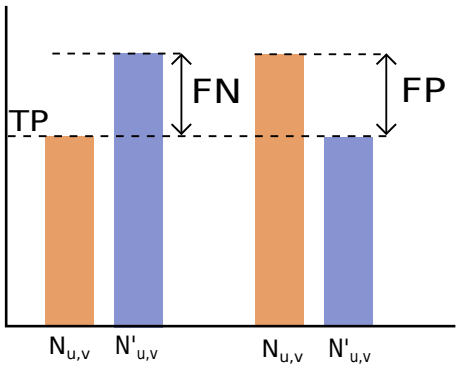

Figure 2: Evaluation scores for activity prediction in link streams.

5 Experiments

In this section, we evaluate the performance of our framework using two datasets which are both real-world contact data between individuals, captured with sensors. We investigate different metrics combinations as well as the influence of the observation period duration. We aim to identify the strengths of our protocol as well as the main challenges to improve the prediction to guide our future works.

5.1 Data description

The first trace used was collected in a French high school in 2012 (*High-school* dataset), see [4] for full details. It is a link stream of 181 nodes and 45047 links, connecting 2220 distinct pairs of nodes over a period of 729,500 seconds (approximately 8 days). Each undirected link (t, u, v) means that the sensor carried by individual u or v detected the sensor carried by the other individual at time t, which means in turn that these two individuals were close enough at time t for the detection to happen. We call this a contact between individuals u and v.

The second dataset has been collected during the IEEE INFOCOM 2006 in Barcelona (*Infocom* dataset) – see [5]. The bluetooth devices used in this experiment recorded connexions with one another. This dataset contains 98 nodes and 283,100 links. During this 3 days long experiment, 4338 pairs of nodes have interacted.

We can see that the *Infocom* dataset involves less nodes but contains more links and more active pairs of nodes. Therefore, comparing between these two datasets allows to get insights about how the density of interconnections affects the performance of the combination of features used.

5.2 Experimental implementation

Our experimental protocol focuses on the performance gain that can be achieved by combining link stream metrics. The dataset is divided into substreams according to the description given in Section 3, with $T = [A, \Omega]$, as an observation stream to predict $L' = (T', V, E')$, where $T' = [A', \Omega']$ with $\Omega = A'$. Then, to understand the information brought by each metric, we combine two metrics at a time, the weight of each metric being related to the parameter α, according to the following equation:

$$f_{u,v}(t) = \alpha \cdot \mathcal{A}_{u,v}(t) + (1 - \alpha) \cdot f'_{u,v}(t) \tag{4}$$

where $\mathcal{A}_{u,v}$ is the function associated to the *pair activity extrapolation* (see 4.1.2), which is considered here as a benchmark to compare with the performance our combination method, $f'_{u,v}$ is a function corresponding to one of the other metrics presented.

5.3 Combinations of temporal and structural features

We study how the use of different metrics with different weights affects the prediction on our real world datasets. For this purpose, we first combine one of the structural functions presented in section 4.1.1 to the function $\mathcal{A}_{u,v}$.

5.3.1 Metrics analysis

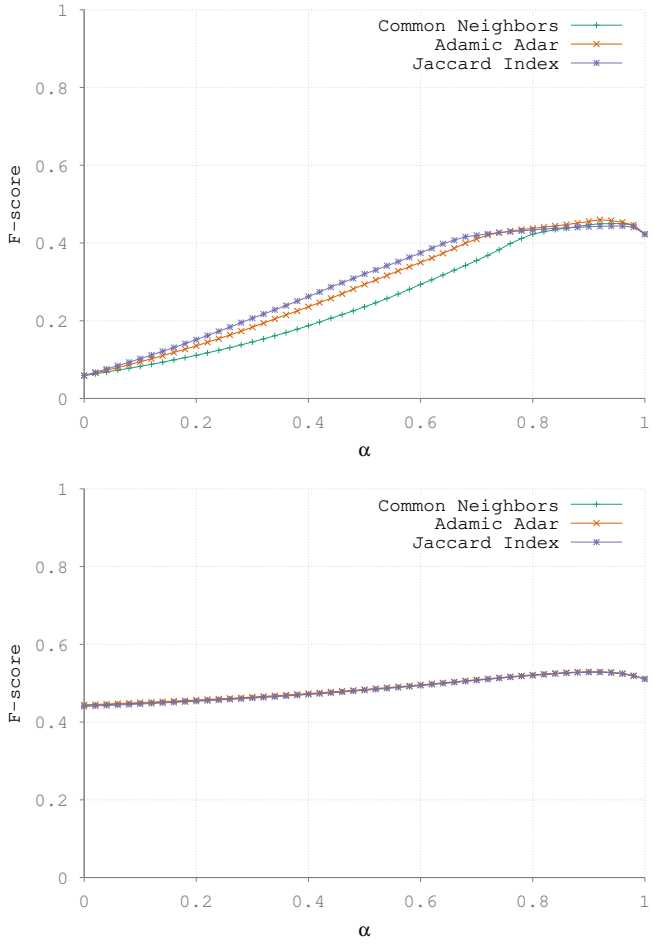

Figure 3: F-score of the predictions with different weight ratios between the *pair activity extrapolation* and a structural metric for the *Highschool* (top) and *Infocom* (bottom) dataset.

Each plot in Figure 3 represents the F-score for a different pair of metrics combination as a function of α for each of the two datasets. Higher values of α represent a greater weight of the pair activity extrapolation in the prediction. We combine successively the *pair activity extrapolation* with the

number of common neighbors, the *Adamic-Adar index* and the *Jaccard index*. Using the *Highschool* intercontact dataset, we build the observation stream using the links that appear between 7:30 am to 8:00 am on the first day. We then predict the link activity between each pair of nodes from 8:00 am to 1:00 pm and compute the prediction performance. The three plots have a similar shape: the F-score values steadily increase with α until reaching a maximum for $\alpha \in [0.90; 0.98]$, and then decrease until $\alpha = 1$. Note that for $\alpha = 1$, the prediction functions come down exactly to the *pair activity extrapolation*, therefore all three functions yield the same F-score.

We then apply the same protocol to the *Infocom* dataset. The observation stream is built using the link appearing between 1:30pm and 2:30pm to predict the links between 2:30pm and 6:00pm. We can see that the three structural metrics tested behave qualitatively in a similar way as in the *Highschool* case, however the values for $\alpha = 0$ are higher, suggesting that the structural metrics alone perform better on the *Infocom* dataset than they do on the *Highschool* one, which is probably due to the higher overall activity of nodes in the former one. We also observe a maximum for values of $\alpha \simeq 0.9$ for each plots with F-score values of 0.52.

These observations indicate that in both experiments, combining a temporal metric with a structural metric may lead to an improvement of the F-score. In these experiments the improvements remain relatively small, from 4.9% to 8.8%, because of the simplicity of the metrics chosen, but it shows that our protocol is able to draw benefit from the combination of temporal and structural information.

5.3.2 Categories analysis

To understand how each type of information affects our prediction, we refine our analysis, without modifying the prediction protocol, by dividing the set of pairs of nodes in two categories. On the one hand, some pairs have not interacted during T, so that when predicting the occurrence of a link in one of them we predict a new link in the stream. We call new link any (t, u, v) in the prediction stream L' such that $\forall x \in T, \nexists\, (x, u, v) \in E$ On the other hand, other pairs have interacted during T and predicting the corresponding link occurrence is predicting link repetition. We call recurrent link any (t, u, v) in the prediction stream L' such that there exist a link $(x, u, v) \in E$. The evaluation method is then applied on the complete set of pairs and on each of these two subsets. We focus on the combination of the *number of common neighbors* with the *pair activity extrapolation*.

We exhibit in Figure 4 the F-score as a function of α for the three categories of pairs aforementioned on both datasets. We can see on the *Highschool* dataset that the F-score corresponding to the recurrent link category increases to a maximum for $\alpha = 0.8$, while for the new link category it remains nearly constant until $\alpha = 0.78$, at which point it grows to a maximum

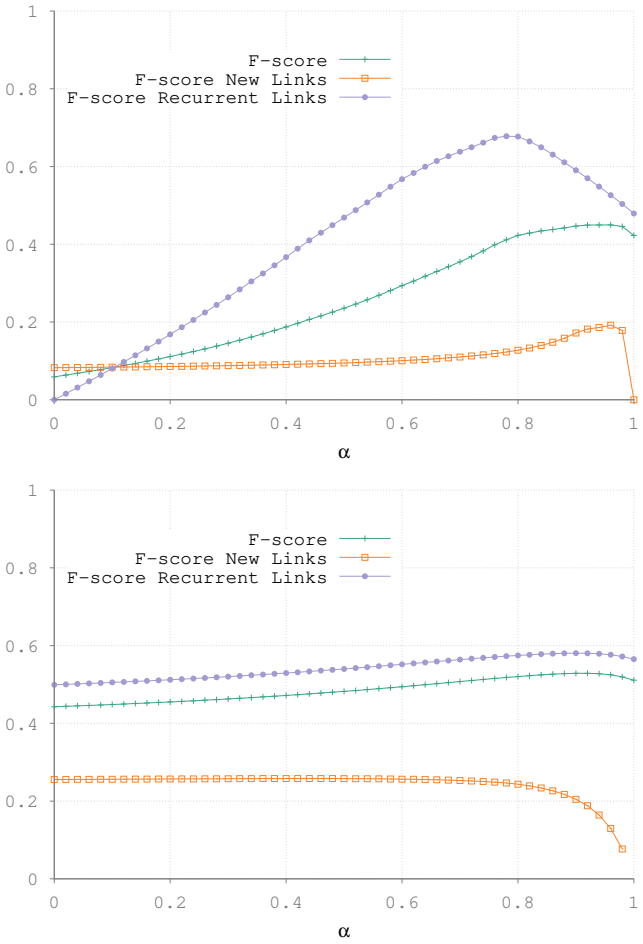

Figure 4: F-score of the predictions between the *pair activity extrapolation* and the *number of common neighbors* as a function of α for different categories of pairs of nodes for the *Highschool* (top) and *Infocom* (bottom) dataset. Green: All links, Orange: New links, Purple: Recurrent inks

reached for $\alpha = 0.96$.

Considering new links, the *pair activity extrapolation* alone is not able to predict this kind of interactions, and thus yields a null F-score. However, we can see that the F-score associated with the new links is almost constant for a wide range of α when the influence of the *number of common neighbors* is predominant in the prediction. This behavior is due to the low number of new links appearing during T': As α grows, less and less links are attributed to these pairs. However, as the number of new links actually appearing is much smaller than the total number of predicted links, both the numbers of true positive and false negative remain almost constant. Therefore, the

recall is also nearly constant. The precision slightly improves as α grows as the number of false positives decreases. When the number of true and false positive are close, it leads to an increase of the F-score corresponding to the prediction of new links for values of $\alpha \simeq 0.9$. Similarly, the performance of the prediction of recurrent links improves as more weight is given to the extrapolation of previous pair activity. However, as the number of links appearing between the pairs of nodes of this category is more significant, we do not see the same effect of stagnation for a wide range of α.

We can observe that the F-score remains relatively low in this type of predictions. The difficulty of this task is mainly due to the class imbalance problem [13]. Given the small number of links occurring compared to the number of pairs of nodes considered, it is a known difficulty in many real world datasets. As expected, this appears more clearly when predicting new links due to the larger number of pairs involved compared to the recurring links.

Note also that the positions of the peaks seem to be closely related to the number of links predicted, it suggests that a possible way to improve the performance of the prediction could be to refine the hypothesis that the activity in L' is the same as the activity in L.

The plot corresponding to the *Infocom* dataset show quite a different kind of behavior, which is surprising as Figure 3 seemed to show similar qualitative behaviors between both datasets. We can see that the F-score accounting for new links prediction starts from 0.25 for $\alpha = 0$ and slowly decreases until $\alpha = 0.8$ at which point it sharply decreases to 0. Regarding the recurrent links prediction the F-score starts from 0.49 and reaches a maximum of 0.57 for $\alpha = 0.9$ and then decreases to 0.56.

This observation may be related to the fact that the *Infocom* dataset is strongly interconnected, with numerous links and active pairs. Therefore, the common neighbors of a given pair of node have often already engaged in repeated interactions, allowing the common neighbors metric to perform well in predicting these links. Concerning the new links prediction we can see that as α grows the increase in precision almost counterbalance the decrease in recall, leading to a slow F-score decay until $\alpha = 0.8$. As the recall value gets closer to 0, the F-score rapidly decreases too. The cause of this behavior is unclear. This may be linked to the fact that the number of common neighbors allows to predict both the new and recurrent links. Therefore we do not see a clear change in the balance between predicting new links and recurrent links as in the *Highschool* dataset.

These experiments highlight the fact that the metric combination does not have the same impact depending on the dataset considered. While each metric tends to predict preferentially a specific type of activity on the *Highschool* dataset, this is not the case in the *Infocom* dataset, where our structural metric is able to predict both new and recurrent links. It also points out the fact that, by choosing specific metrics combination, the

prediction can be focused on different kinds of activity, involving different kinds of links.

5.4 Time intervals variations

We investigate here how the variation of the observation duration affects the performance of the protocol. Figure 5 shows the F-score when using different observation periods on the two datasets. We use the same experimental protocol, combining the *pair activity extrapolation* with the *number of common neighbors*. Concerning the *Highschool* dataset the prediction period starts at 10:30am and ends at 3:30pm. We then apply our method for different observation periods. The observation period duration is successively 30, 60, 90 and 150 minutes ending at 10:30am.

When α is close to 1, the prediction yields higher F-score for longer periods of observation times than for shorter ones. As the *pair activity extrapolation* is the main contributor to the prediction function, longer periods lead to a better averaging of the activity between each nodes. However, considering observation periods from 30 to 90 minutes, the F-score presents a maximum for $\alpha < 1$, while the one associated with observation duration of 150 minutes is maximum for $\alpha = 1$.
It is interesting to note that the maximum associated with an observation period of 90 minutes is really close to the maximum score obtained for 150 minutes, meaning that combining different information allowed us to make prediction of similar efficiency using a shorter observation period.

We then applied the same protocol on the *Infocom* dataset, the prediction going from 2:30pm to 6:00pm. We used the same observation period duration as before. We can see that the plots are qualitatively similar to the ones that be obtained on the *Highschool* dataset, except for the fact that the plot associated to the longest observation period displays a small peak at $\alpha = 0.98$. It also appears that when α is small our protocol performs better for shorter observation periods, while longer observation periods lead to better score, when α is close to 1.

In these experiments, we observe that mixing structural and temporal information improves the prediction on short observation periods but not necessarily on long ones. The boundary between what can be considered as a short or a long observation period is of course a disputable matter, and largely depends on the dataset under study. This is due to the fact that when the observation period is longer, links that have not appeared in the past are less likely to appear in the dataset and recurrent links are therefore predominant. Thus, extrapolating the previous stream activity is more relevant than using structural information to predict the future activity of a pair of nodes. We think that in datasets which exhibit a growing activity rate, the structural information would be able to help predict new links even when using long observation periods.

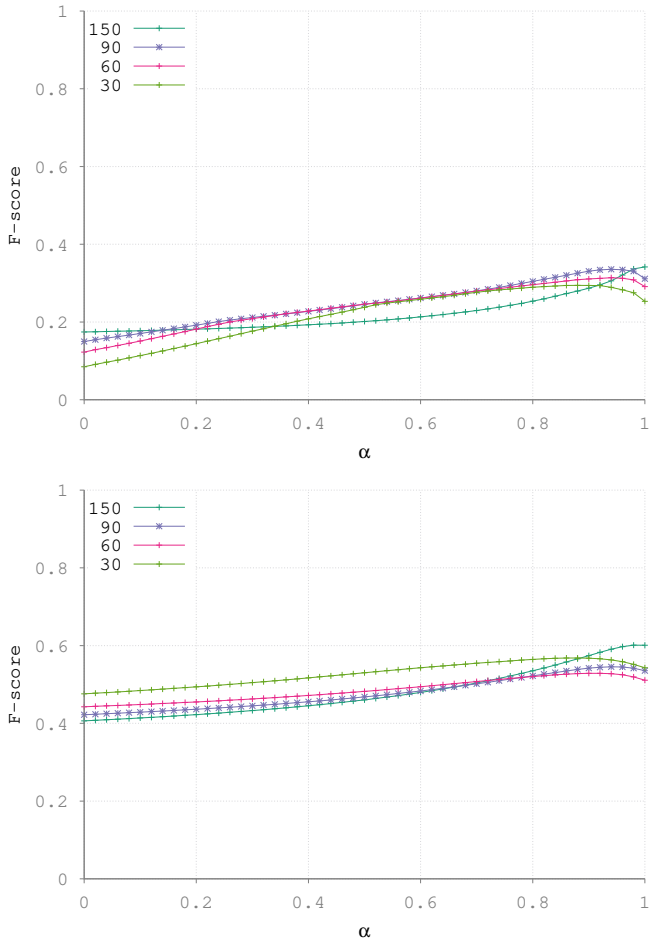

Figure 5: F-score of the predictions combining the *pair activity extrapolation* and the *number of common neighbors*, as a function of α for different observation durations, for the *Highschool* (top) and *Infocom* (bottom) datasets.

5.5 Temporal features combination

We now investigate how our protocol performs when using more advanced temporal metrics. In this series of experiments, we combine the *pair activity extrapolation* with one of the temporal metrics presented in 4.1.2 with the same protocol. We used $\mathcal{A}_k$ with $k = 5$, meaning that we compute the activity from the last 5 links before Ω, $\mathcal{A}_\delta$ with $\delta = 500$ seconds (the activity is computed from the last 500 seconds before Ω), and $\mathcal{A}_n(t)$ with $n = 10$ (the activity is computed from a linear extrapolation over 10 points of the observation period. We applied these metrics to both datasets and increased the duration of the observation period to study our metrics behavior when

the dynamics of the system change over time.

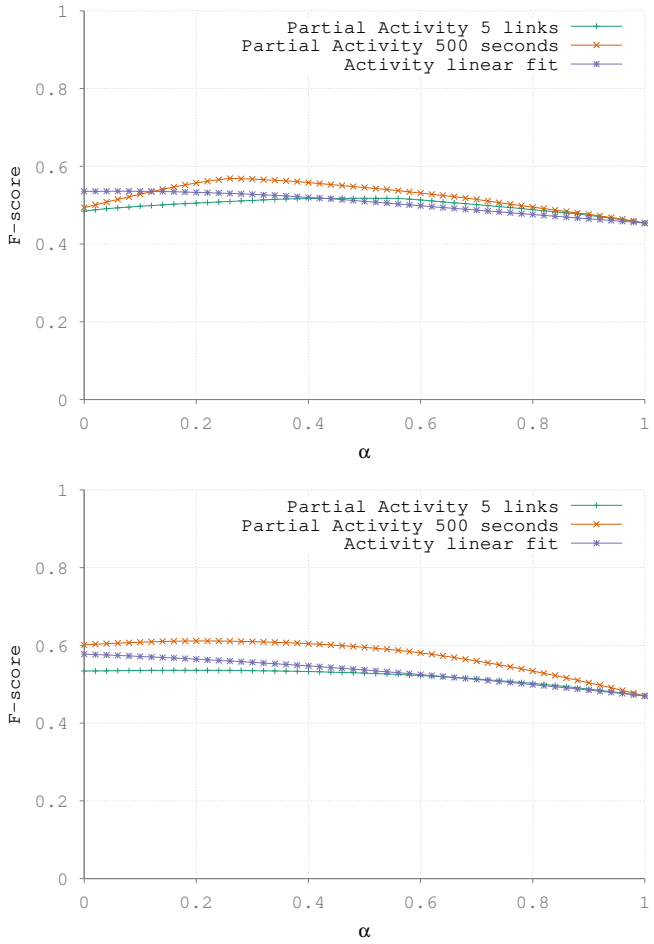

Figure 6: F-score values for the *Highschool* (top) and *Infocom* (bottom) dataset. Each plot represents the F-score of a prediction using the *pair activity extrapolation* and other temporal metrics.

Concerning the *Highschool* dataset the observation period is set from 6:30am to 8:00am and the prediction period from 8:00am to 8:30am. We can see in Figure 6 (top) that the F-score when using $\mathcal{A}_{k=5}$ starts from 0.48, then grows to 0.51 for $\alpha = 0.5$ and then decreases to 0.45 for $\alpha = 1$. The F-score related to $\mathcal{A}_{\delta=500}$ has a similar shape with a maximum of 0.56 for $\alpha = 0.26$ When using $\mathcal{A}_{n=10}(t)$, we observe a linear decrease as α grows, from a 0.53 F-score to 0.45.

For the *Infocom* dataset we set the observation from 6:00am to 2:00pm and the prediction from 2:00pm to 4:00pm. In Figure 6 (bottom), we can see that our metrics perform in a similar way as what we saw with the

Highschool dataset.

In both cases, we observe that the two partial activity extrapolation metrics ($\mathcal{A}_\delta$ and $\mathcal{A}_k$) provide information complementary to the benchmark, as they perform better when combined with it. This can be interpreted as a better balance between the short and long term dynamics. On the other hand, the activity fit seems to be systematically an improvement to our benchmark but the prediction does not benefit from the combination. This tells us that combining different temporal metrics can improve our prediction performance. It also shows that combining a variety of temporal metrics focusing on specific dynamical properties allows to control the weight of each of these properties in the prediction.

6 Conclusion

In this work, we proposed an activity prediction protocol adapted to the link stream formalism, making it possible to advantageously use the rich information contained in this modeling. It is built around a flexible way to combine the information from metrics which capture features of the stream. We also proposed an evaluation protocol adapted to our problem. Our experiments show that combining structural and temporal features leads to performance improvements. We also showed that the length of the observation period have complex consequences on the prediction, that demands to be studied in depth. This work is a first step towards activity prediction in link streams. Our protocol is designed in a modular way, such that each part is independent from the others and can be replaced or improved, depending on the application we are interested in.

Therefore, different improvements are considered for future works. The metrics presented in this work are classical metrics used for link prediction in graphs or basic ways to capture the temporal information of the stream. As our protocol is ready to combine new metrics, we intend to design refined measurements that are able to detect more subtle dynamical features of the stream, *e.g.* we expect that giving weight to recent links would enhance the prediction. We also consider implementing pattern mining techniques to identify typical motifs of the short term dynamics. For example, we could consider that if three nodes u, v, w occasionally interact with each other by short bursts of activity, the occurrence of links between the pairs u, v and u, w suggest a link apparition between v and w shortly after. Finally, we made the assumption that the activity remains constant from the observation period to the prediction period. However, this hypothesis is not always satisfied and greatly depends on the data under consideration. Models developed in the context of time series prediction, like the ARIMA model which extrapolates precisely the past activity [15], would certainly allow to better evaluate the number of links predicted.

Acknowledgement

This work is funded in part by the European Commission H2020 FET-PROACT 2016-2017 program under grant 732942 (ODYCCEUS), by the ANR (French National Agency of Research) under grants ANR-15-CE38-0001 (AlgoDiv) and ANR-13-CORD-0017-01 (CODDDE), by the French program "PIA - Usages, services et contenus innovants" under grant O18062-44430 (REQUEST), and by the Ile-de-France program FUI21 under grant 16010629 (iTRAC).

References

[1] T. Viard and M. Latapy, "Identifying roles in an ip network with temporal and structural density," in *Conference on Computer Communications Workshops (INFOCOM WKSHPS)*. IEEE, 2014, pp. 801–806.

[2] P. Holme and J. Saramäki, "Temporal networks," *Physics reports*, vol. 519, no. 3, pp. 97–125, 2012.

[3] A. Casteigts, P. Flocchini, W. Quattrociocchi, and N. Santoro, "Time-varying graphs and dynamic networks," *International Journal of Parallel, Emergent and Distributed Systems*, vol. 27, no. 5, pp. 387–408, 2012.

[4] R. Mastrandrea, J. Fournet, and A. Barrat, "Contact patterns in a high school: a comparison between data collected using wearable sensors, contact diaries and friendship surveys," *PloS one*, vol. 10, no. 9, p. e0136497, 2015.

[5] J. Scott, R. Gass, J. Crowcroft, P. Hui, C. Diot, and A. Chaintreau, "CRAWDAD dataset cambridge/haggle (v. 2009-05-29)," Downloaded from http://crawdad.org/cambridge/haggle/20090529, May 2009.

[6] D. Liben-Nowell and J. Kleinberg, "The link-prediction problem for social networks," *Journal of the American society for information science and technology*, vol. 58, no. 7, pp. 1019–1031, 2007.

[7] Z. Huang, X. Li, and H. Chen, "Link prediction approach to collaborative filtering," in *Proceedings of the 5th ACM/IEEE-CS joint conference on Digital libraries*. ACM, 2005, pp. 141–142.

[8] M. Al Hasan, V. Chaoji, S. Salem, and M. Zaki, "Link prediction using supervised learning," in *SDM06: workshop on link analysis, counter-terrorism and security*, 2006.

[9] L. Lü and T. Zhou, "Link prediction in complex networks: A survey," *Physica A: Statistical Mechanics and its Applications*, vol. 390, no. 6, pp. 1150–1170, 2011.

[10] G. Kossinets, "Effects of missing data in social networks," *Social networks*, vol. 28, no. 3, pp. 247–268, 2006.

[11] L. Tabourier, A.-S. Libert, and R. Lambiotte, "Predicting links in ego-networks using temporal information," *EPJ Data Science*, vol. 5, no. 1, p. 1, 2016.

[12] M. Pujari and R. Kanawati, "Supervised rank aggregation approach for link prediction in complex networks," in *Proceedings of the 21st International Conference on World Wide Web*. ACM, 2012, pp. 1189–1196.

[13] R. N. Lichtenwalter, J. T. Lussier, and N. V. Chawla, "New perspectives and methods in link prediction," in *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2010, pp. 243–252.

[14] T. Tylenda, R. Angelova, and S. Bedathur, "Towards time-aware link prediction in evolving social networks," in *Proceedings of the 3rd workshop on social network mining and analysis*. ACM, 2009, p. 9.

[15] Z. Huang and D. K. Lin, "The time-series link prediction problem with applications in communication surveillance," *INFORMS Journal on Computing*, vol. 21, no. 2, pp. 286–303, 2009.

[16] L. A. Adamic and E. Adar, "Friends and neighbors on the web," *Social networks*, vol. 25, no. 3, pp. 211–230, 2003.