Interaction Prediction Problems in Link Streams

Thibaud Arnoux, Lionel Tabourier, Matthieu Latapy

Interaction Prediction Problems in Link Streams

Thibaud Arnoux, Lionel Tabourier and Matthieu Latapy

1 Introduction

Analyzing interactions over time plays a key role in many contexts: recommender systems (who buys which product and when), contacts between individuals (message exchanges, physical proximity or phone calls, for instance), and transaction analysis (like money or data transfers) are typical examples. As a consequence, much effort is devoted to the analysis of such data with approaches like temporal networks, time-varying graphs or link streams [4, 2, 6].

Predicting future interactions is a crucial question in all these contexts, but the problem is traditionally addressed by merging interactions into a graph or series of graphs, called snapshots [7, 9, 12]. This has the key advantage of building a bridge with the powerful formalism and tools of graph theory, but at the cost of important information losses. More importantly, we argue that this approach misses interesting variants of the problem itself.

The goal of this chapter is to deepen our understanding of these interaction prediction problems. To do so, we formalize them within the link stream framework, which makes it possible to fully capture both the temporal and structural nature of data. This leads to several meaningful problem definitions, that raise quite different challenges, as well as relations between them and classical approaches.

We focus here on problem definitions and comparisons; resolving some of them has already received attention [5, 3, 1] but unifying them into the same framework leads to a better understanding of the whole and the identification of new variants of interest. We also show that this helps to identify general approaches to tackle them.

Throughout this chapter, we assume a standard approach for solving prediction problems. First, one designs a model in order to make a prediction based on the fundamental assumption that future behaviors can be predicted from past observations.

Thibaud Arnoux, Lionel Tabourier, Matthieu Latapy
Laboratoire d'Informatique de Paris 6, Sorbonne Université, CNRS, UMR7606, F-75005 Paris, France
e-mail: first_name.last_name@lip6.fr

Then, parameters of the model are learned from past data, using an optimization process which aims at maximizing the prediction quality. Therefore, each prediction problem demands several ingredients, among which a quality estimator, features to describe past data and a model to combine these features.

We first introduce the data modeling with link streams,which is the framework that we choose to address the interaction prediction problems. Afterward we present the prediction problems themselves, classified with respect to their ambition in the prediction task and we also discuss the subtle question of prediction evaluation. Finally, we propose a general direction for solving these problems using what we call *pairwise likeliness functions*.

2 Link stream modeling of interactions

We use here the instantaneous link stream formalism presented in [6], which is a special case of stream graphs where nodes are always present and links have no duration. Such a link stream L is defined as a triplet (T,V,E) where $T = [\alpha,\omega] \subseteq \mathbb{R}$ is a time interval, V is the set of nodes under concern and $E \subseteq T \times V \otimes V$ is a set of links: $(t,uv) \in E$ means that u et v interacted at time t. We consider here undirected interactions between pairs of distinct nodes u and v, which we denote by $uv \in V \otimes V$. We assume that E is finite: it contains a finite number of interactions, each occurring between two distinct nodes at a specific time instant. We illustrate this modeling in Figure 1. Extending our work to more general cases is left for future work.

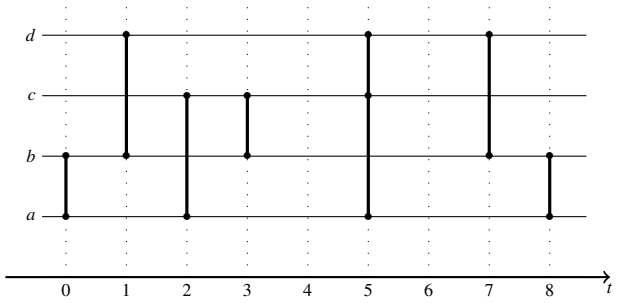

Fig. 1 An example of undirected instantaneous link stream like the ones considered in this chapter: $L = (T,V,E)$ with $T = [0,8]$, $V = \{a,b,c,d\}$ and $E = \{(0,ab),(1,bd),(2,ac),(3,bc),(5,ac),(5,cd),(7,bd),(8,ab)\}$.

Such a link stream $L = (T,V,E)$ naturally induces a graph $G = (V,E')$ defined by $E' = \{uv : \exists t,(t,uv) \in E\}$: it is the graph in which two nodes of V are linked together if they interacted at some time in T. Dually, the link stream induces a time series $(\ell_t)_{t\in T}$ defined by $\ell_t = |\{uv : (t,uv) \in E\}|$: ℓ_t is the number of interactions occurring at time t.

In this context, the classical link prediction problem in graphs consists in predicting from G the links that will appear in the future, and the classical time series prediction problem consists in predicting from $(\ell_t)_{t\in T}$ the number of interactions that will appear in the future. Our aim here is to draw benefit from L to predict richer information on future interactions. Depending on the targeted information, this leads to several, quite different problems, that we detail in the next section.

3 Prediction problems and evaluation

Throughout the rest of this chapter, we assume that the set of nodes V remains unchanged, in other words nodes do not appear nor disappear. All the prediction problems that we consider start with an *input stream* $L_i = (T_i, V, E_i)$ with $T_i = [\alpha_i, \omega_i]$ and $E_i \subseteq T_i \times V \otimes V$. A prediction is related to an *output stream* $L_o = (T_o, V, E_o)$ with $T_o = [\alpha_o, \omega_o]$ and $E_o \subseteq T_o \times V \otimes V$. The time interval T_o is called the *prediction period*. Interactions actually occur during this period; we model them as the *actual stream* $L_a = (T_a, V, E_a)$ with $T_a = T_o$ and $E_a \subseteq T_a \times V \otimes V$. In addition, we always assume here that $\omega_i \leq \alpha_o$; in other words we focus on predicting *future* interactions. Within this framework, the prediction is considered as good if the properties of L_o are similar to those of L_a.

3.1 Predicting all interactions

3.1.1 Description

Predicting all interactions of all pairs in the actual stream may be the most ambitious formulation of the problem. It means that we aim at predicting each appearing link, i.e. predicting the stream L_a. We represent in Figure 2 the situation corresponding to a given prediction.

3.1.2 Quality evaluation

To evaluate the quality of such a prediction, a measurement of the distance between L_a and L_o is necessary. For a given pair of nodes, the series of actual or predicted interactions between them comes down to a set of points in $T_a = T_o$. Consequently, one may use a distance between two sets of points to evaluate the distance between the streams, and thus the prediction quality.

Among possible choices, let us mention the nearest point distance: the distance from a point $x \in X$ to the set Y is the distance from x to the closest point in Y, and the distance from X to Y is the sum of the distances of each $x \in X$ to Y. Though simple, this measurement is not formally a distance as it is not symmetric. Therefore, we

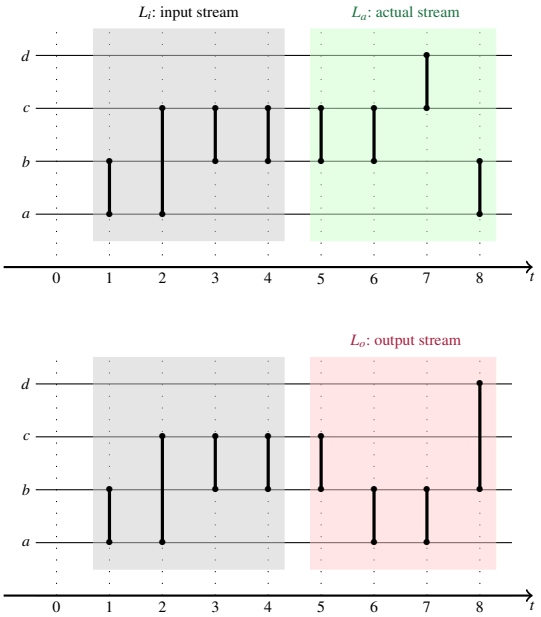

Fig. 2 Top: schematic representation of the input stream L_i and the corresponding actual stream L_a. Bottom: schematic representation of the input stream L_i and the corresponding output stream L_o. The problem of predicting each interaction leads to comparing L_a to L_o.

may use instead the spike train distance proposed by Victor and Purpura [14], which was originally designed to evaluate how different two neuronal impulse trains are. We suggest to define the distance between link streams as

$$D(L_o, L_a) = \sum_{uv \in V \otimes V} D^{uv}(L_o, L_a)$$

where $D^{uv}(L_o, L_a)$ is the spike train distance between the points representing the interactions between u and v in L_o and L_a. $D^{uv}(L_o, L_a)$ is defined as the minimal cost to transform one set of points into the other with elementary steps: either deleting, adding points or moving points along the time axis. Finally, $D(L_o, L_a)$ can be understood as the minimal cost to transform L_o into L_a with these elementary steps. When attributing a fixed cost to the addition and deletion steps, it is a metric distance (see [14] for more details). To give the reader a more precise idea of the meaning of this distance without diving in too much technical details, we represent in Figure 3 an example of minimal cost transformation of a set of time points into another.

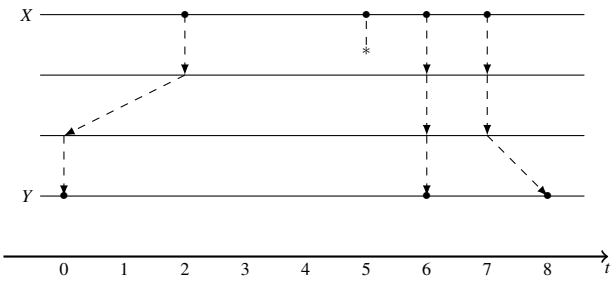

Fig. 3 Minimal cost transformation of a set of time points X into another Y: the first step is deleting a point from X (cost = fixed penalty), the other steps consist in translating points along the time axis (cost = translation time distance).

3.2 Predicting the next interaction for each pair of nodes

3.2.1 Description

A less constrained version of the former prediction task consists in predicting only the next interaction for each pair of nodes (if it exists). Indeed, in many contexts an experimenter is mostly interested in the moment when the next interaction occurs, as represented in Figure 4. For example, when predicting interactions in order to spread an information through a network, the experimenter is interested in knowing when the next interaction happens to spread the message as soon as possible. This task has the advantage to circumvent the difficult prediction of the number of links per pair of nodes.

In this case, the output of the prediction is not a stream, but the next occurrence time for each pair $uv \in V \otimes V$. In order to include the case where there is no interaction between u and v during the time interval of prediction T_o, a legitimate definition for the object predicted is the set $\{t_o^{uv}\}_{uv \in V \otimes V}$, with $t_o^{uv} \in [\alpha_o, \omega_o] \cup \{\infty\}$, with $t_o^{uv} = \infty$ meaning that we predict no interaction for uv.

3.2.2 Quality evaluation

In terms of quality evaluation, we should quantify the difference between the sets $\mathscr{T}_o = \{t_o^{uv}\}_{uv \in V \otimes V}$ predicted and $\mathscr{T}_a = \{t_a^{uv}\}_{uv \in V \otimes V}$ actually occurring. Point set distances such as the ones proposed in the previous task can be used here too, and they are simpler with this prediction task, considering the fact that we take into account at most one interaction for each stream.

We denote d a distance function between two points in time. Then a possible distance which can be seen as an equivalent of the spike train distance in this simpler case is

$$D(\mathscr{T}_a, \mathscr{T}_o) = \sum_{uv \in V \otimes V} d(t_a^{uv}, t_o^{uv}) = \sum_{uv \in V \otimes V} min(|t_a^{uv} - t_o^{uv}|, p)$$

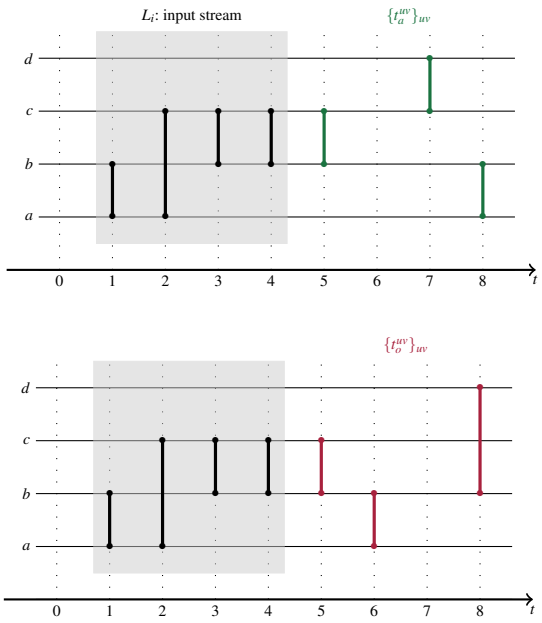

Fig. 4 Top: schematic representation of the input stream L_i and the corresponding actual set of next occurrence times $\{t_a^{uv}\}_{uv}$. Bottom: schematic representation of the input stream L_i and the corresponding output set of next occurrence times $\{t_o^{uv}\}_{uv}$. The problem of predicting the next interaction for each pair of nodes leads to comparing $\{t_a^{uv}\}_{uv}$ to $\{t_o^{uv}\}_{uv}$.

Using $d(t_a^{uv}, t_o^{uv}) = min(|t_a^{uv} - t_o^{uv}|, p)$ means that the distance between t_a^{uv} and t_o^{uv} is either the delay between these interaction times, or a predefined penalty p if there is no interaction between u and v in L_a and we predicted one in L_o (and *vice versa*). Thus it is similar to the addition/deletion cost of the spike train distance mentioned in Sec. 3.1. Here again, this quality evaluation is a simple and natural choice from our point of view, but other choices are available.

With this evaluation, the distance depends linearly on the time gap between a predicted link and a link observed in the actual stream. However, a user might consider that a linear dependence is not appropriate to describe the problem accurately and that other functions might be more relevant. In Figure 5, we represent the case of a sigmoid-like distance function of the time gap. This distance function is complementary to a similarity function $s(x,y)$ such that $d(x,y) = 1 - s(x,y)$ with $y < x$, which is represented on the figure.

According to the evaluation method described above, we interpret the quality of the prediction using a notion of temporal distance between two events. Another possible interpretation of this evaluation method consists in using the vocabulary of classification tasks, as what is done in the case of link prediction problems. Indeed, if a link uv is observed in the actual stream at instant t_a^{uv} while it is not predicted yet,

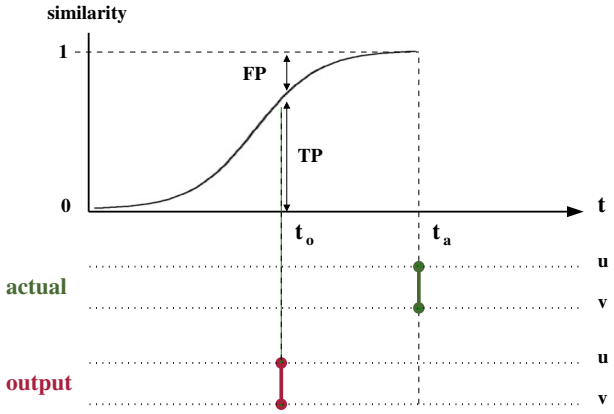

Fig. 5 Representation of the level of similarity between a predicted and an actual link, and its interpretation in terms of true and false positive prediction. In the situation represented, the interaction is predicted at instant t_o, which occurs before the actual interaction at t_a, the prediction is thus both a true positive and a false positive to a certain extent. This extent is computed by a sigmoid-like similarity function.

it can be interpreted as an equivalent of a *false negative*. Conversely, a link which is predicted while it is not observed yet is a *false positive*.

Of course, a link almost never occurs at the exact time when it has been predicted. Consequently, it is desirable not to use a 0/1 notion of false positive or false negative, but rather a score in the interval $[0,1]$ which accounts for how close or how far we are from an exact prediction. That is what the similarity function s defined above does. To describe in more details the parallel between the classification-based to the distance-based interpretations, $s(t_a^{uv},t_o^{uv})$ with $t_a^{uv} > t_o^{uv}$ quantifies the similarity, i.e. the degree of correctness of the prediction, while $1-s(t_a^{uv},t_o^{uv})$ represents the degree of error as a *false positive FP* does. If $t_o^{uv} > t_a^{uv}$, $1-s(t_o^{uv},t_a^{uv})$ rather represents the degree of error as a *false negative FN* prediction does. The degree of correctness can be mapped to the notion of *true positive TP*, which is consistent with the fact that $s(t_a^{uv},t_o^{uv}) = 1$ when the link has been predicted exactly at the right time. Using this framework of interpretation, an unpredicted link is equivalent to a link predicted at $t_o^{uv} = \infty$, and similarly a non-occurring link is equivalent to a link occurring at $t_a^{uv} = \infty$.

It should be noted that *TP*, *FP*, *FN* are usually boolean values which are defined unambiguously, while here the result depends on the choice of the distance function d. Besides that, *true negative (TN)* predictions do not have any obvious equivalent using temporal distances. However, it is enlightening to interpret the prediction with both the vocabulary of classification and temporal distances.

3.3 Predicting the number of interactions for each pair of nodes

3.3.1 Description

Rather than predicting *if and when* each pair interacts, another relevant task consists in predicting *how many times* each pair interacts in a given period. It is less ambitious than the previous tasks, in the sense that we do not request to predict the exact time of links occurrence. In this case, the temporal precision of the prediction only depends on the duration of the output stream, and as it can be adjusted in the prediction protocol, we can tune how precise the prediction is in regards to the temporal dimension.

To formalize more precisely the prediction task, we define the notion of *activity of a pair of nodes uv* in the stream $L = (T,V,E)$ as $\mathscr{A}^{uv} = |\{(t,uv) \in E\}|$. In this context, our goal is that for any $uv \in V \otimes V$, $\mathscr{A}_o^{uv} = \mathscr{A}_a^{uv}$. Note that the activity quantifies the multiplicity of interactions between two nodes, so it is often represented by the weight of a link in the graph formalism.

3.3.2 Quality evaluation and relation to the link prediction problem

In this case, distance measures such as the train spike distance cannot be used directly, as we do not predict interaction times. This task is actually closer to a more usual link prediction task on a graph snapshot, where the snapshot length corresponds to the duration of the actual stream, and we can draw advantage from that. We design quality estimators in the same spirit as what has been done in Sec. 3.2.2, by defining equivalents to *TP*, *FN* or *FP* predictions.

As *FP* correspond to events that do not happen but are predicted, it is legitimate to translate this idea as the difference between the number of predicted links and the number of actual links if the former is larger than the later. Similarly, *FN* correspond to events which occur but are not predicted, so it translates to the opposite difference if there are more actual links than there are predicted links. *TP* are the events which occur and are predicted so it is equivalent to the minimum between these two activities. Formally:

$$\begin{cases} |TP^{uv}| = \min(\mathscr{A}_a^{uv}, \mathscr{A}_o^{uv}) \\ |FP^{uv}| = \max(\mathscr{A}_o^{uv} - \mathscr{A}_a^{uv}, 0) \\ |FN^{uv}| = \max(\mathscr{A}_a^{uv} - \mathscr{A}_o^{uv}, 0) \end{cases}$$

These definitions are illustrated in Figure 6.

The definitions of *true positive*, *false positive* and *false negative* proposed here satisfy usual relationships concerning these indicators: $|TP^{uv}| + |FP^{uv}|$ is the number of predicted interactions, $|TP^{uv}| + |FN^{uv}|$ is the number of interactions between *u* and *v* in the actual stream.

Then, we denote $|TP|$, (resp. $|FN|$, $|FP|$) the total number of true positive (resp. false negative, false positive) in the stream:

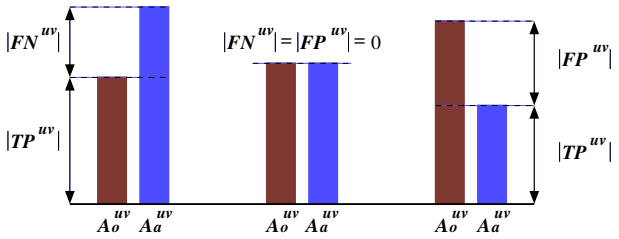

Fig. 6 Illustrations of the three possible cases ($\mathscr{A}_a^{uv} > \mathscr{A}_o^{uv}$, $\mathscr{A}_a^{uv} = \mathscr{A}_o^{uv}$ and $\mathscr{A}_a^{uv} < \mathscr{A}_o^{uv}$) of the interpretations of *TP*, *FN* or *FP* in the context of the prediction of the number of interactions for each pair of nodes.

$$\begin{cases} |TP| = \sum_{uv \in V \otimes V} |TP^{uv}| \\ |FN| = \sum_{uv \in V \otimes V} |FN^{uv}| \\ |FP| = \sum_{uv \in V \otimes V} |FP^{uv}| \end{cases}$$

We can thus define accordingly useful quantities to evaluate the quality of a prediction:

- precision: $\frac{|TP|}{|TP|+|FP|}$, which represents the fraction of good predictions among the total number of predictions,
- recall: $\frac{|TP|}{|TP|+|FN|}$, which represents the fraction of events detected among the total number of events which can be detected,
- F1-score, which is the harmonic mean of the precision and recall, that is to say $F_1 = 2 \cdot \frac{precision \cdot recall}{precision + recall}$.

Using this interpretation, a good prediction can be considered for example as a prediction that maximizes the F1-score, as it reflects a compromise between precision and recall. Nevertheless, we do not define any equivalent to a *true negative* prediction and to the total number of negative predictions in general. It makes us unable to define equivalents of other classification estimators (fall-out, specificity, ROC curve, etc.).

As stated previously, this problem relates to the weighted link prediction problem: given a weighted graph representing the number of interactions between each pair of nodes, predict the future weight. Related problems exist in the link prediction literature. For instance, some authors have proposed to divide links into two families: new links and recurring links, and then make two separate predictions for each family [13]. Besides that, our task can also be related to the matrix completion problem, which is usually considered with boolean adjacency matrices in the context of link prediction (see for example [10]) but can be generalized to matrices with positive values. Powerful as they may be, these approaches leave in the shadow the fundamentally temporal nature of the data, which our formulation of the problem tries to grasp.

3.4 Predicting the existence of interaction(s) for each pair of nodes

Finally, a natural problem is predicting if a pair interacts at least once in the actual stream. Another way of formulating this task using the activity defined in Section 3.3, could be to predict for all pairs of nodes if they reach an activity of 1 during the prediction period. An interesting point concerning this prediction task is that it is actually similar to the classical link prediction problem: the prediction task comes to predicting the structure of the actual graph $G_a = (V, E'_a)$ aggregated from L_a, where uv is in E'_a if there is at least one (t, uv) in L_a. The main difference is that the link stream formalism stresses the fact that both structural and temporal information can be used as features to improve the prediction quality. Such information has already been used in the literature in order to achieve link prediction tasks, but in a more classical framework (see e.g., [8, 9]).

In terms of evaluation, link prediction tasks have been widely studied as binary classification tasks, and thus one makes use of the evaluators usually employed for such issues (precision, recall, F-scores, ROC curve, etc.).

4 Pairwise likeliness functions for prediction tasks

From now on, we suppose that the prediction problem and its evaluation method are set, and we focus on the prediction model. We present in this section a possible way to address these prediction problems taking into account the fact that the data contains structural and temporal information. Consequently, the features of the stream that we use for prediction should be described in a way that can reflect both structural and temporal properties.

The overall approach is the following. First, we compute pairwise likeliness functions using properties of the input stream L_i. Pairwise likeliness functions are designed to reflect when we expect a pair of nodes to interact during the period T_o. The prediction model relies on these pairwise likeliness functions: one would train the parameters of the model by maximizing the quality of the prediction on a learning stream, L_a using the vocabulary defined in Section 3. After this learning phase, the model can be used for prediction.

4.1 Pairwise likeliness functions

In order to represent a feature of the input stream on which the prediction is based, we use a function $f^{uv}(t)$ which represents the likeliness for a link uv to occur at time t. An interesting aspect of this approach is that it gathers in a same formalism both temporal and structural (and potentially hybrid) features. We call such functions *pairwise likeliness functions*.

4.1.1 Illustration

To set the reader's mind, we illustrate this notion using three examples. Let us consider the following features:

1) a structural feature often used in link prediction problems: the *number of common neighbors* shared by two nodes,
2) a temporal feature based on the assumption that there is some regularity in the temporal patterns of interaction between two nodes, that we call *regularity*,
3) another temporal feature which is used to reflect the fact that there are episodes of bursty activity of interactions, the *burstiness*.

For these examples, we suggest possible definitions of the corresponding pairwise likeliness functions. These definitions are based on common sense, but other possibilities would make sense. Our goal here is to show that this formalism is versatile.

1) Concerning the number of common neighbors, the pairwise likeliness function is a constant (independent from time), which is simply the number of common neighbors itself

$$f_{CN}^{uv}(t) = |\{w : \exists\ (t_1, uw), (t_2, wv) \in E_i\}|$$

2) Regularity is defined using the interaction times between u and v during the input stream. Supposing that the links are approximately regularly spaced, a consistent shape for the likeliness function could be a sinusoidal function, as sketched in Figure 7. The corresponding definition is

$$f_{reg}^{uv}(t) = \frac{1}{2} + \frac{1}{2} cos\left(\frac{2\pi(t - t_\ell)}{\langle\tau\rangle}\right)$$

where t_ℓ denotes the last interaction time of uv in L_i and $\langle\tau\rangle$ the average interaction time during T_i.

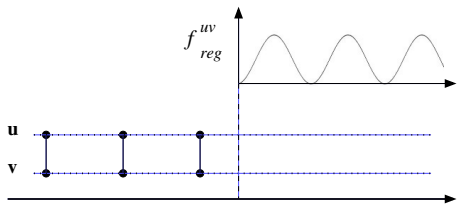

Fig. 7 Representation of a sinusoidal regularity-based likeliness function. Bottom: input stream. Top: corresponding regularity-based likeliness function computed from the input.

3) Finally concerning burstiness, we consider that if a train of interactions (that is to say more than two interactions) has begun less than a time δ ago, then there is

an increased probability of interaction during the next δ duration, as represented in Figure 8. The corresponding definition is

$$f^{uv}_{burst}(t) = \begin{cases} 1 \text{ if } t \in [0;\delta] \text{ and } |\{(t,uv) \in E_i \text{ with } t \in [-\delta;0]\}| > 2 \\ 0 \text{ else.} \end{cases}$$

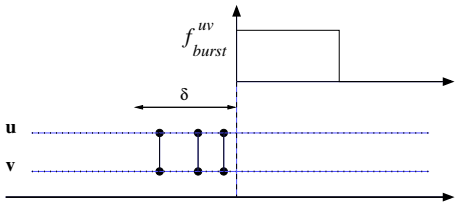

Fig. 8 Representation of a rectangular burstiness-based likeliness function. Bottom: input stream. Top: corresponding burstiness-based likeliness function computed from the input.

4.1.2 Combining pairwise likeliness functions

In order to achieve the prediction itself, we now define a prediction model based on pairwise likeliness functions. We combine these functions into $\mathscr{F}^{uv}(t)$, the *combined pairwise likeliness function*. Again, there are many possible ways to achieve this combination, we propose to use a linear combination as an illustration of the approach:

$$\mathscr{F}^{uv}(t) = a_{CN} \cdot f^{uv}_{CN}(t) + a_{reg} \cdot f^{uv}_{reg}(t) + a_{burst} \cdot f^{uv}_{burst}(t)$$

In this framework, the coefficients represent the weight given to the different features in the combination. On the examples of the three pairwise functions previously defined, the combination for one pair uv is represented graphically in Figure 9.

4.2 Combined pairwise likeliness functions for prediction tasks

Now that, for each pair of nodes, we have a function representing the likeliness of an interaction during the prediction period, we discuss how this function can be used to achieve the prediction tasks formerly presented.

We have seen in Section 3 that there are two different kinds of tasks: On the one hand, predicting the appearance of one or several links, that is to say predicting precisely the triplets (t,uv) (tasks 1 and 2); on the other hand, predicting the number of links which occur during a given period of time (tasks 3 and 4). We discuss these two families of prediction tasks separately.

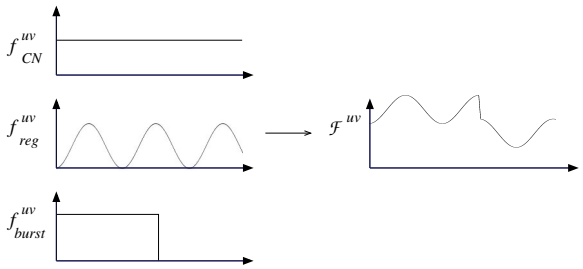

Fig. 9 Illustration of a combined pairwise likeliness function for a pair of nodes uv, based on the linear combination of f_{CN}, f_{reg} and f_{burst}.

4.2.1 Predicting one or several link occurrences

Given a pair of nodes, the goal is to predict what are the occurrence times – if any – of interactions between these nodes. A natural way could be to detect the local maximum of the combined likeliness function. In this case, the problem seems to map to detecting peaks in a function equivalent to a time-series. A given point of a time series is said to be a peak if the associated value is larger than a specified threshold. Peak detection is an active area of research, and techniques could be derived from this field (see for example [11]). A problem is that such methods aim at identifying points which stand out from their neighbors in the times series, while here a user would rather consider that a long plateau should correspond to the existence of one or even several interactions. In other words, the problem is not exactly equivalent to the intuition of a peak detection method. It may be closer to the burst detection problem (e.g, [15]): one looks for a time-window during which the aggregated signal is larger than a user-specified threshold. However, burst detection usually focuses on locating a period of high activity, rather than a precise point in time. Both tasks are thus not identical in that case too.

In any case, there should be additional criteria to consider if a peak is significant enough to justify the prediction of an interaction. One way of doing so is to define an area under the curve around the peak, and the peak is considered significant if this area is larger than a threshold, as schematically represented on Figure 10. Formally, if a peak has been detected at time τ, the criterion of significance would be $\int_{\tau-\delta}^{\tau+\delta} \mathscr{F}^{uv}(t)\,\mathrm{d}t \geq \Theta$, where Θ is the significance threshold of the area around the peak, and δ characterizes the width defining the area under the curve around the peak. The choices of δ and Θ are obviously critical for the prediction task, as it will largely influence the number of links in the predicted stream. The parameters of any method should thus be carefully chosen based on the input stream.

Note that predicting only the next interaction does not alleviate the problem of peak significance mentioned above and also calls for non-trivial choices to decide if a peak is significant enough to justify the existence of an interaction.

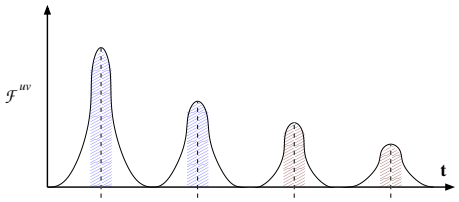

Fig. 10 Illustration of a criterion to select significant peaks: the areas colored are the area under $\mathscr{F}^{uv}$ centered on a local maximum and of width 2δ. Areas in blue are larger than a threshold Θ, while areas in brown are smaller than Θ.

Another issue should be mentioned when several link occurrences are predicted. Interactions are not independent from each other, meaning that if an interaction is predicted at time t, it should affect the probability for an interaction to occur at any time $t' > t$. This issue is simply ignored when the problem is managed as a peak detection problem, which is another limitation to this technique. To address the issue, it is possible to predict interactions sequentially, first considering the next occurrence, then assuming it does happen in order to predict the next one, etc. However, other difficulties appear when tackling the problem along these lines, one of which being the accumulation of prediction errors throughout the process.

4.2.2 Predicting the number of interactions over a given period

When considering the prediction of a number of links during a given period, one would certainly use the likeliness functions differently. As we no longer predict the interaction occurrence times, it is not necessary to detect the peaks of the likeliness function.

The area under the likeliness function curve represents the likeliness for a link to appear over the whole prediction period. Therefore, one could consider that this area should be related to the number of interactions actually occurring during that period. So, supposing that we are able to predict efficiently the total number of interactions in the prediction stream, it is possible to predict the number of links for any pair uv by allocating links to pairs of nodes proportionally to the area under the likeliness function curve. In short, a relevant relation for predicting the activity of a pair uv over the prediction period $T_o = T_a = [\alpha_o, \omega_o]$ is to consider that

$$\mathscr{A}_o^{uv} = C \cdot \int_{\alpha_o}^{\omega_o} \mathscr{F}^{uv}(t)\,\mathrm{d}t$$

where C is a constant fixed by the total number of interactions in the output stream. As predicting this number is a classical time-series prediction task, we have in our hands the tools to achieve the prediction of the number of links for any pair of nodes in the stream during a given period.

5 Conclusion

In this chapter, we have formulated the problem of predicting interactions in a link stream, which can be seen as a generalization of the link prediction problem in a network when the temporal dimension of the data is taken into account, or dually, as the generalization of a time series prediction, when there is a network-like structure supporting the various time-series.

We have seen that the most general problems, predicting exactly the moments when two nodes in the stream will interact with each other, is certainly a difficult task to achieve – as could be expected. But we have also proposed different, more humble tasks, which seem simpler to address as they are closer to more classical prediction problems, namely the link prediction task in a graph. Precisely, the task of predicting the number of links which appear during a given period of time seems promising. Indeed, it allows to use evaluation metrics which can be interpreted to a certain extent using the vocabulary of classification tasks, and we presented a possible way to tackle this prediction using features of the input stream that would account for its structural and temporal characteristics.

We do not develop in this chapter the details of the technical implementation of this method. However, an interested reader can go to [1] for a more comprehensive view of an implementation on contact networks, which suggests that there are indeed good prospects (and still a lot to do) on these prediction tasks.

References

1. Thibaud Arnoux, Lionel Tabourier, and Matthieu Latapy. Predicting interactions between individuals with structural and dynamical information. *arXiv preprint arXiv:1804.01465*, March 2018.
2. Arnaud Casteigts, Paola Flocchini, Walter Quattrociocchi, and Nicola Santoro. Time-varying graphs and dynamic networks. *International Journal of Parallel, Emergent and Distributed Systems*, 27(5):387–408, 2012.
3. Paulo Ricardo da Silva Soares and Ricardo Bastos Cavalcante Prudêncio. Time series based link prediction. In *Neural Networks (IJCNN), The 2012 International Joint Conference on*, pages 1–7. IEEE, 2012.
4. Petter Holme and Jari Saramäki. Temporal networks. *Physics reports*, 519(3):97–125, 2012.
5. Zan Huang and Dennis KJ Lin. The time-series link prediction problem with applications in communication surveillance. *INFORMS Journal on Computing*, 21(2):286–303, 2009.
6. Matthieu Latapy, Tiphaine Viard, and Clémence Magnien. Stream graphs and link streams for the modeling of interactions over time. *arXiv preprint arXiv:1710.04073*, 2017.
7. David Liben-Nowell and Jon Kleinberg. The link-prediction problem for social networks. *Journal of the Association for Information Science and Technology*, 58(7):1019–1031, 2007.
8. Ryan N Lichtenwalter, Jake T Lussier, and Nitesh V Chawla. New perspectives and methods in link prediction. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 243–252. ACM, 2010.
9. Linyuan Lü and Tao Zhou. Link prediction in complex networks: A survey. *Physica A: Statistical Mechanics and its Applications*, 390(6):1150–1170, 2011.

10. Feiping Nie, Hua Wang, Xiao Cai, Heng Huang, and Chris Ding. Robust matrix completion via joint schatten p-norm and lp-norm minimization. In *Data Mining (ICDM), 2012 IEEE 12th International Conference on*, pages 566–574. IEEE, 2012.
11. Girish Palshikar et al. Simple algorithms for peak detection in time-series. In *Proc. 1st Int. Conf. Advanced Data Analysis, Business Analytics and Intelligence*, pages 1–13, 2009.
12. Purnamrita Sarkar, Deepayan Chakrabarti, and Michael Jordan. Nonparametric link prediction in dynamic networks. *arXiv preprint arXiv:1206.6394*, 2012.
13. Christoph Scholz, Martin Atzmueller, and Gerd Stumme. On the predictability of human contacts: Influence factors and the strength of stronger ties. In *Privacy, Security, Risk and Trust (PASSAT), 2012 International Conference on and 2012 International Confernece on Social Computing (SocialCom)*, pages 312–321. IEEE, 2012.
14. Jonathan D Victor and Keith P Purpura. Metric-space analysis of spike trains: theory, algorithms and application. *Network: computation in neural systems*, 8(2):127–164, 1997.
15. Yunyue Zhu and Dennis Shasha. Efficient elastic burst detection in data streams. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 336–345. ACM, 2003.