PAPER

Link Prediction in Social Networks Using Information Flow via Active Links

Lankeshwara MUNASINGHE[†a)], *Nonmember and* **Ryutaro ICHISE**[†,††b)], *Member*

SUMMARY Link prediction in social networks, such as friendship networks and coauthorship networks, has recently attracted a great deal of attention. There have been numerous attempts to address the problem of link prediction through diverse approaches. In the present paper, we focused on predicting links in social networks using information flow via active links. The information flow heavily depends on link activeness. The links become active if the interactions happen frequently and recently with respect to the current time. The time stamps of the interactions or links provide vital information for determining the activeness of the links. In the present paper, we introduced a new algorithm, referred to as *T_Flow*, that captures the important aspects of information flow via active links in social networks. We tested *T_Flow* with two social network data sets, namely, a data set extracted from Facebook friendship network and a coauthorship network data set extracted from *ePrint archives*. We compare the link prediction performances of *T_Flow* with the previous method *PropFlow*. The results of *T_Flow* method revealed a notable improvement in link prediction for facebook data and significant improvement in link prediction for coauthorship data.

key words: *link prediction, time stamps, link activeness, social networks*

1. Introduction

Link prediction was introduced as a way to infer which new links are likely to occur in the near future in a given network [9]. If we are presented with a snapshot of a network at the current time, the goal is to predict links that will occur in the future. The structural information, features of nodes and edges of the given network can be used to predict future links.

Link prediction has many applications and, it offers many benefits to the users of social networking services. For example, online social networking services, such as Facebook, can use link prediction to provide their users with better recommendations or suggestions. Therefore, users of these services can efficiently find their friends, colleagues, or people whom they wish to meet. Organizations such as research organizations, business organizations, and security agencies will be able to uncover information regarding unseen relationships among people or organizations. Thus, they may operate more effectively. Link prediction in scientific researcher networks allow researchers to find experts and research organizations in the same research field [22]. However, highly structured massive real-world networks involving heterogeneous entities with complex associations have added new challenges to link prediction research. Supervised and unsupervised learning methods have been used in previous studies with different frameworks for link prediction [4], [8]. The machine learning approaches remain an immense challenge due to different factors such as sparsity, complexity, size, time-dependent nature of the networks and imbalance between possible links and actual links observed in these networks [10].

Information flow between nodes is a vital factor for link evolution in social networks. It varies over time depending on the activeness of the links between nodes. It is worthwhile to study the factors which determine the information flow and how these factors can be effectively used for link prediction in networks. Particularly, the activeness of links is one of the key factors which determines the information flow. Some of the recent link prediction research have introduced supervised/unsupervised methods based on information flow in social networks. One of them is *PropFlow* algorithm [10]. This algorithm has used random walk to determine the information flow between nodes. Link weights are the transition probabilities for the random walker. If a node pair has higher transition probability, more information flow happens between the node pair and the node pair is more likely to get linked in the future. One supervised random walk algorithm [2] learns link strengths using link and node attributes and uses the strengths as the transition probabilities. However, those studies haven't been considered the activeness of the links. We therefore, introduced a new algorithm which considers the effect of information flow via active links for link evolution.

The remainder of the present paper is organized as follows. Section 2 discuss related research in link prediction in social networks and biological networks. The newly proposed algorithm, *T_Flow* has been introduced in Sect. 3. Experimental evaluation and experimental results are presented in Sect. 4. Section 5 presents our conclusions.

2. Related Work

In this section, we review some of the research related to link prediction as well as background information on link prediction. The increase in the number of studies related to link prediction in the recent past reveals the emerging interest in link prediction. Diverse approaches, including machine learning approaches and probabilistic approaches,

Manuscript received November 7, 2012.
Manuscript revised March 6, 2013.

[†]The authors are with the Graduate University for Advanced Studies, Tokyo, 101–8430 Japan.
[††]The author is with National Institute of Informatics, Tokyo, 101–8430 Japan.
a) E-mail: lankesh@nii.ac.jp
b) E-mail: ichise@nii.ac.jp

DOI: 10.1587/transinf.E96.D.1495

have been proposed in order to address the problem of link prediction.

Classification using a learned model is the prominent feature of machine learning. Supervised and unsupervised machine learning methods have been widely used for link prediction in coauthorship networks [16] along with set of structural features of networks introduced in [9]. Later, the introduction of new features such as cooccurrence probability [21], keyword match count for paper topics and abstracts [18] in combination with supervised machine learning methods provided more accurate link predictions in coauthorship networks. The supervised learning approach introduced for predicting link strengths using transactional information by [6] shows the correlation between varying link strength and future link evolution. These previous studies have proven the consistency and effectiveness of machine learning methods in link prediction.

Besides machine learning approaches, there are different approaches can be seen in the literature. Parametric probabilistic model based on topological features of networks has introduced in [7] for link prediction in biological networks. A matrix alignment method was used to determine the most predictive features of coauthorship networks by aligning adjacency matrix of a network with weighted similarity matrices [19]. The weighted similarity matrices are computed from node attributes and neighborhood topological features. The weights learned by minimizing an objective function.

The recent research [12] has introduced a new feature which captures the impact of information flow via active links for link evolution in social networks. However, it is limited to common neighbors. We therefore, introduced *T_Flow* algorithm which computes the information flow between any pair of nodes in a social network by considering the link activeness. Once we compute it, we used it as a feature for link prediction using supervised machine learning methods.

3. Supervised Learning Method for Link Prediction

Most of the approaches discussed in the previous section have used structural features of networks and the features of the nodes and edges for link prediction. For example, the features such as number of common neighbors, Jaccard's coefficient [11] are used to measure the similarity between nodes. Once these features are computed for a particular node pair, we obtain a vector of values referred to as a *feature vector* [16], which may be correlated with the future possible link between that node pair.

In supervised learning approach, we train the learning system with the feature vectors of each node pair to learn a model which can be used to predict the future links. Once we compute the feature vectors for each node pair in a network, we obtain a set of feature vectors for node pairs that are already linked and another set of feature vectors for node pairs that are not linked. The learning system is trained to learn a model using the feature vectors and the model used to predict unlinked node pairs that are to be linked in the future.

3.1 Features Used for Link Prediction

Table 1 lists the details of the features used in the present study. We used two different combinations of features in the proposed machine learning approach for link prediction. The two sets of features includes a set of features used in [12] with *PropFlow* score computed by previous *PropFlow* algorithm [10] and *T_Flow* score computed by *T_Flow* algorithm introduced in this paper. One set was used as the *PropFlow combination* which includes the *PropFlow* score and used as the base line combination. The other set is the *T_Flow combination*, which includes the *T_Flow* score introduced herein.

The existing features used in [12] are described below.

Adamic/Adar [1] This measure indicates if a node pair has a common neighbor which is not common to several other nodes, then the similarity of that particular node pair is higher than the node pairs having neighbors that are common to several other nodes. This measure assigns higher weights to common neighbors that are not common to several other nodes.

Common neighbors Number of common neighbors of a node pair.

Jaccard's coefficient [11] Normalized measure of common neighbors.

Preferential attachment [14] This measure indicates that new links are more likely to be formed with nodes of higher degree, or nodes that are popular in the network.

We have shown the feature computation formulas in Table 1 for a pair of nodes i and j. In the formulas, $\Gamma(i)$ and $\Gamma(j)$ denote the sets of neighbors of i and j respectively, k denote a node. In Sect. 3.2, we discuss the computation method of the previous algorithm *PropFlow* and in Sect. 3.3 we discuss the computation method of the new algorithm *T_Flow* introduced in this paper.

Table 1 Feature listing.

Feature	Formula	PropFlow combination (PFC)	T_Flow combination (TFC)
Adamic/Adar	$\sum_{k \epsilon \Gamma(i) \bigcap \Gamma(j)} \frac{1}{log\lvert\Gamma(k)\rvert}$	✓	✓
Common neighbors	$\lvert\Gamma(i) \bigcap \Gamma(j)\rvert$	✓	✓
Jaccard's coefficient	$\frac{\lvert\Gamma(i) \bigcap \Gamma(j)\rvert}{\lvert\Gamma(i) \bigcup \Gamma(j)\rvert}$	✓	✓
Preferential attachment	$\lvert\Gamma(i)\rvert\lvert\Gamma(j)\rvert$	✓	✓
PropFlow		✓	-
T_Flow		-	✓

3.2 PropFlow Algorithm

Information flow between nodes is a vital factor for link evolution in social networks and it depends very much on link attributes such as link weights and activeness. The *PropFlow* algorithm [10] focused on information flow. It computes information flow based on random walk method which select its path based on link weights. This method is somewhat similar to rooted page rank, but restricted to local neighborhood of a node. Unlike rooted page rank, the random walker doesn't need to restart or convergence and use modified breadth first search restricted to depth l. The random walker starts from a particular node and reach the desired node in l steps or fewer. Revisiting any node including starting node is not allowed for the random walker. *PropFlow* algorithm computes the information flow called *PropFlow* for a pair of nodes i and j based on the random walks between them. Equation (1) shows how to compute $PropFlow(i, j)$ if nodes i and j directly linked. In this case, random walker starts from node i and walk to node j.

$$PropFlow(i, j) = NodeInput_i * \frac{w_{ij}}{\sum_{k \in N(i)} w_{ik}} \tag{1}$$

Where, w_{ij} denotes the weight of the link between nodes i and j. k denotes a node and set $N(i)$ denotes node i's neighbors whose depth is greater than the depth of node of i from the starting node. Initial node input is regarded as 1. If nodes i and j are indirectly linked, *PropFlow* algorithm computes the information flow through all the shortest paths from node i to node j using Eq. (1) recursively and take the summation.

For example, Eqs. (2) to (7) show how to compute the $PropFlow(A, D)$ between nodes A and D in the coauthorship network shown in Fig. 1. Link weights are denoted by p. We assumed the random walker starts from node A. $PropFlow(A, D)$ is computed using link weights of links AB, BC, CD, BE, ED. There are four paths the random walker can reach node D from node A. They are A→B→C→D, A→B→E→D, A→B→E→C→D, and A→B→C→E→D. We have to note that *PropFlow* algorithm use modified breadth-first search method and it stops when revisiting any node. Thus, random walker doesn't revisit node C from node E. Therefore, the paths A→B→E→C→D and A→B→C→E→D have not considered for computations. First, we have to compute $PropFlow(A, B)$. Weight of the link between A and B is 3. The sum of the link weights of links between A and its neighbors is 4. Note that initial node input of A is considered as 1. $PropFlow(A, B)$ can be compute as;

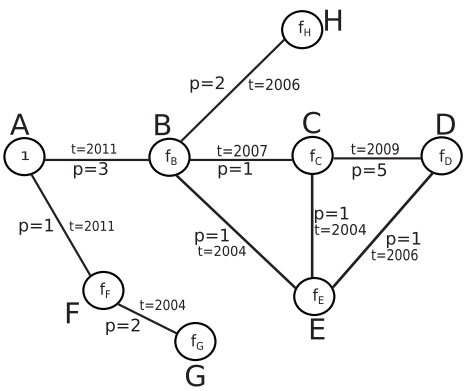


Fig. 1 An example of a coauthorship network.

$$PropFlow(A, B) = 1 * \frac{3}{(1+3)} = 1 * \frac{3}{4} = \frac{3}{4} \tag{2}$$

$PropFlow(B, C)$ can be compute as;

$$PropFlow(B, C) = PropFlow(A, B) * \frac{1}{(1+1+2)} = \frac{3}{4} * \frac{1}{4} = \frac{3}{16} \tag{3}$$

$PropFlow(B, E)$ can be compute as;

$$PropFlow(B, E) = PropFlow(A, B) * \frac{1}{(1+1+2)} = \frac{3}{4} * \frac{1}{4} = \frac{3}{16} \tag{4}$$

$PropFlow(C, D)$ can be compute as;

$$PropFlow(C, D) = PropFlow(B, C) * \frac{5}{5} = \frac{3}{16} * 1 = \frac{3}{16} \tag{5}$$

$PropFlow(E, D)$ can be compute as;

$$PropFlow(E, D) = PropFlow(B, E) * \frac{1}{1} = \frac{3}{16} * 1 = \frac{3}{16} \tag{6}$$

Therefore, the $PropFlow(A, D)$ is;

$$PropFlow(A, D) = \frac{3}{16} + \frac{3}{16} = \frac{6}{16} = \frac{3}{8} \tag{7}$$

Although *PropFlow* algorithm computes the information flow in social networks using link weights, the information flow doesn't depend only on the link weights. The activeness of the links is a vital factor for information flow. The links become weak or deactivate if nodes haven't interacted recently with respect to the current time. Despite of their weights, the weakened or deactivated links can cause a decay in information flow. We therefore, introduced an extension of *PropFlow* algorithm referred to as *T_Flow* algorithm in order to consider the effect of active links for information flow.

3.3 T_Flow Algorithm

The time stamps of the links or interactions are useful in determining the activeness of the links. If a node pair interact recently the link between them become active. In

other words, the time stamp of the last interaction is a vital information in deciding the activeness of a link. Hence, we used the most recent time stamps of the interactions between nodes for our computations. Time stamp can be the most recent hour, day or year of a interaction between a node pair. The time unit of the time stamps depends on the network. *T_Flow* algorithm use the same settings as in *PropFlow* algorithm for random walk. It considers link weight as well as link activeness to compute transition probabilities. The *T_Flow* algorithm computes the information flow called *T_Flow* between a pair of nodes in a network. We assumed the decay of information flow as a function of decaying factor α and difference of time stamps of adjacent links. The decaying function $d(i, j)$ for information flow from node i to its adjacent node j is defined as;

$$d(i, j) = (1 - \alpha)^{|t_x - t_y|} \tag{8}$$

The decaying factor α ($0 < \alpha < 1$) is the rate of decay per unit time of the information flow and t_x is the time stamp of the link which random walker comes into the node i and t_y is the time stamp of the link which random walker going to node j. The value of decaying function become 1 when $\alpha = 0$ which means no decay in information flow. At this point *T_Flow* algorithm is identical to its previous version *PropFlow* algorithm. The *T_Flow* algorithm computes the information flow from node i to j via direct link as follows;

$$T\_Flow(i, j) = NodeInput_i * \frac{w_{ij}}{\sum_{k \in N(i)} w_{ik}} * (1 - \alpha)^{|t_x - t_y|} \tag{9}$$

If nodes i and j are indirectly linked, *T_Flow* algorithm computes the information flow through all the shortest paths from node i to node j using Eq. (9) recursively and take the summation. The total flow between two nodes regarded as the *T_Flow* for the node pair. At the start of the random walk, t_x is regarded as the current time and the initial node input is considered as 1. We have listed the *T_Flow* algorithm in Algorithm 1.

For example, Eqs. (10) to (15) show how to compute $T\_Flow(A, D)$ between nodes A and D in Fig. 1. Time stamps of the links denoted by t in Fig. 1. We assumed the random walker starts from node A and the current time is the year 2012. $T\_Flow(A, D)$ is computed using link weights of links AB, BC, CD, BE, ED and their time stamps. First, we have to compute $T\_Flow(A, B)$.

$$\begin{aligned} T\_Flow(A, B) &= 1 * \frac{3}{(1 + 3)} * (1 - \alpha)^{|2012-2011|} \\ &= \frac{3}{4} * (1 - \alpha)^1 = \frac{3}{4} * (1 - \alpha) \end{aligned} \tag{10}$$

The link BC has the time stamp (2007) and the link BE has the time stamp (2004). Therefore, BC is the most active link. Thus, more information should flow through BC than BE which has the same weight as BC but less active than BC. $T\_Flow(B, C)$ can be compute as;

Algorithm 1: T_Flow Algorithm

```
Input: network G = (V, E), start node s, depth l, decaying factor
       α, current time t_c
Output: T_Flow T_f for all neighbors of s within depth l
begin
    insert s into Visitedset
    push s into NewSearchqueue
    push t_c into Timequeue
    insert (s,1) into T_f
    OldSearchqueue ⟵ empty
    for Distance← 0 to l do
        OldSearchqueue ⟵ NewSearchqueue
        empty NewSearchqueue
        while OldSearchqueue is not empty do
            pop i from OldSearchqueue
            pop t_x from Timequeue
            find NodeInput using i in T_f
            t_y ← 0
            SumWeight ← 0
            Flow ← 0
            for j in neighborhood of i do
                if depth of j > depth of i then
                    add weight of edge between i and j to
                    SumWeight
                end
            end
            for j in neighborhood of i do
                if depth of j > depth of i then
                    w_ij ← weight of edge between i and j
                    t_y ← time stamp of edge between i and j
                    Flow ←
                    NodeInput * (w_ij / SumWeight) * (1 − α)^|t_x − t_y|
                    add (j, Flow) into T_f
                    if j is not in Visitedset then
                        insert j into Visitedset
                        push j into NewSearchqueue
                        push t_y into Timequeue
                    end
                end
            end
        end
    end
end
```

$$\begin{aligned} &T\_Flow(B, C) \\ &= T\_Flow(A, B) * \frac{1}{(2 + 1 + 1)} * (1 - \alpha)^{|2011-2007|} \\ &= \frac{3}{4} * (1 - \alpha) * \frac{1}{4} * (1 - \alpha)^4 \\ &= \frac{3}{16} * (1 - \alpha)^5 \end{aligned} \tag{11}$$

$T\_Flow(B, E)$ can be compute as;

$$\begin{aligned} &T\_Flow(B, E) \\ &= T\_Flow(A, B) * \frac{1}{(2 + 1 + 1)} * (1 - \alpha)^{|2011-2004|} \\ &= \frac{3}{4} * (1 - \alpha) * \frac{1}{4} * (1 - \alpha)^7 \\ &= \frac{3}{16} * (1 - \alpha)^8 \end{aligned} \tag{12}$$

$T\_Flow(C, D)$ can be compute as;

$$T\_Flow(C,D) = T\_Flow(B,C) * \frac{5}{5} * (1-\alpha)^{|2007-2009|}$$
$$= \frac{3}{16} * (1-\alpha)^7 \quad (13)$$

$T\_Flow(E,D)$ can be compute as;

$$T\_Flow(E,D) = T\_Flow(B,E) * \frac{1}{1} * (1-\alpha)^{|2004-2006|}$$
$$= \frac{3}{16} * (1-\alpha)^{10} \quad (14)$$

Therefore, the $T\_Flow(A,D)$ is;

$$T\_Flow(A,D) = \frac{3}{16} * (1-\alpha)^7 + \frac{3}{16} * (1-\alpha)^{10} \quad (15)$$

4. Experimental Evaluation

At first, we analyzed effectiveness of *T_Flow* algorithm for link prediction by varying α from 0 to 0.9. Then, the link prediction performances of *PropFlow* algorithm and *T_Flow* algorithm were compared using feature combinations *PropFlow combination* which includes *PropFlow* algorithm and *T_Flow combination* which includes *T_Flow* algorithm. For the comparison, we conducted the experiments for *T_Flow combination* using two-loop cross validation where the inner loop determines α and the outer-loop evaluates prediction. Training data in the outer loop is used in the inner loop to find the optimal parameter value which is then used to evaluate the test data in the outer loop. The feature combinations used in the experiments are shown in Table 1. In our experiments, J48 Weka implementation [5] of C4.5 decision tree algorithm [17] was used with 10-fold cross validation. All network data sets are very sparse and hence SMOT oversampling algorithm [3] was used in order to deal with class imbalance problem. Precision, recall and F-measure are used as performance metrics in the experiments. In both *PropFlow* and *T_Flow* algorithms, the depth l is set to 3 which means we excluded the nodes that are more than three links away from a node. We tested the effectiveness of *T_Flow* algorithm for a data set extracted from facebook social network and coauthorship data sets extracted from *e-print archive*[†].

4.1 Experiment with Facebook Data

Facebook data set is a set of wall postings collected from the regional facebook network of New Orleans from September, 2006 to January, 2009 [20]. This data set consist of wall postings exchanged by 60,290 users who are connected by 1,545,686 links. We extracted six different snapshots of data from May, 2008 to December, 2008 which shows a rapid increase of wall postings. Wall postings are considered as the interactions between users. Each data set consist of wall postings of three weeks. Link weight represents the number of wall postings exchanged between a pair of users. The day of the most recent wall posting represents the time stamp of a link.

Table 2 Statistics of Facebook data.

Training data	Nodes	Edges	Clustering coefficient	Mean degree
D1	7094	13294	0.0270	1.87
D2	12862	29656	0.0292	2.30
D3	9310	18138	0.0277	1.94
D4	14405	30142	0.0242	2.09
D5	19614	51030	0.0319	2.60
D6	17277	36414	0.0300	2.10

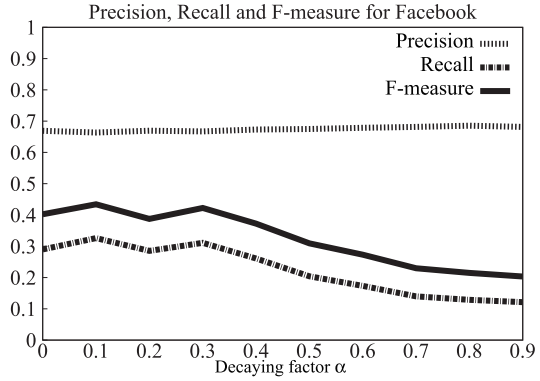


Fig. 2 Performance of *T_Flow combination* for different α values (Facebook data).

We train the decision tree algorithm for Facebook data using wall postings in two consecutive weeks to predict links in the following week. The statistics of the facebook training data are shown in Table 2. The unit of time for Facebook data is days. The experiment was conducted for six data sets and the average performance of *T_Flow* algorithm was computed.

Link prediction performance of *T_Flow combination* with the variation of α for Facebook data is shown in Fig. 2 which was reported in [13]. Average recall and average F-measure shows peaks at $\alpha = 0.1$ and $\alpha = 0.3$ and then decrease as α increase from 0.3 to 0.9 while the average precision doesn't show any drastic changes. We obtained the highest average F-measure for *T_Flow combination* at $\alpha = 0.1$. The decaying factor α measures the decay of influence of wall posting exchange per unit time on information flow. The links become more active if users exchange wall postings frequently and recently. Hence, the information flow decays with the time if users don't exchange wall postings frequently and recently. The facebook network grow rapidly over time and the interactions happen within a quick time. As a consequence, the decay in information flow per unit time (per day) proportionately small. In other words, if a wall posting does not exchange within a day the decay of information flow is proportionately low. Hence, the results are better for the smaller α values (smaller decay). As shown in Table 2, the clustering coefficients and mean degrees of the data is fairly small. It implies that the users interact with less number of friends and only few links are active during the particular time period.

Table 3 shows the comparison of *PropFlow combi-*

[†]http://arxiv.org/

Table 3 Comparison of *PropFlow combination* and *T_Flow combination* for Facebook data.

Feature combination	Avg. Precision	Avg. Recall	Avg. F-measure
PropFlow	**0.6692**	0.2898	0.4023
T_Flow	0.6658	**0.3327**	**0.4412**

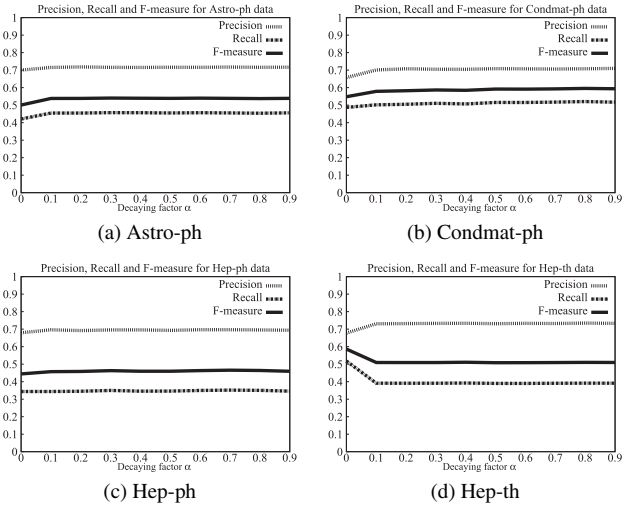
Table 4 Statistics of coauthorship data.

Data set (Subject area)	Training data	Nodes	Edges	Clustering coefficient	Mean degree
Astro-ph	D1 (1992–1996)	8098	53086	0.6974	6.55
	D2 (1993–1997)	12647	113924	0.7092	9.00
	D3 (1994–1998)	17346	177390	0.7062	10.22
	D4 (1995–1999)	22180	261724	0.7042	11.80
	D5 (1996–2000)	27067	358794	0.7031	13.25
	D6 (1997–2001)	31526	455670	0.6992	14.45
Condmat-ph	D1 (1992–1996)	8798	35288	0.6269	4.01
	D2 (1993–1997)	14197	67120	0.6702	4.73
	D3 (1994–1998)	20410	108926	0.6965	5.33
	D4 (1995–1999)	27053	157530	0.7139	5.82
	D5 (1996–2000)	33461	209852	0.7229	6.27
	D6 (1997–2001)	40786	278152	0.7336	6.81
Hep-ph	D1 (1992–1996)	9029	56108	0.5879	6.21
	D2 (1993–1997)	10670	71328	0.6004	6.68
	D3 (1994–1998)	12230	88644	0.6082	7.24
	D4 (1995–1999)	13189	98494	0.6095	7.46
	D5 (1996–2000)	14325	136754	0.6237	9.54
	D6 (1997–2001)	15259	139362	0.6315	9.13
Hep-th	D1 (1992–1996)	8438	24904	0.4904	2.95
	D2 (1993–1997)	9459	29286	0.4976	3.09
	D3 (1994–1998)	10242	33026	0.5094	3.22
	D4 (1995–1999)	10543	35322	0.5164	3.35
	D5 (1996–2000)	11001	38648	0.5146	3.51
	D6 (1997–2001)	11392	41212	0.5162	3.61

nation and *T_Flow combination* for facebook data. We tested *T_Flow combination* using two-loop cross validation method for determining α and 10-fold cross validation for computing the results. The results shows that average F-measure of *T_Flow combination* is better than the average F-measure of *PropFlow combination* which implies that the information flow via active links is a vital factor for link prediction.

4.2 Experiment with Coauthorship Data

The coauthorship data sets extracted from *e-print archive* within ten years period of publications on subject areas Astro physics (Astro-ph), Condensed matter physics (Condmat-ph), High energy physics (theory) (Hep-th) and High energy physics (phenomenology) (Hep-ph) from 1992 to 2002. We created six data sets for each subject area and computed the average performance of *T_Flow* algorithm. We have shown the statistics of each coauthorship network in the Table 4. Publications are considered as the interactions between authors and the year of the most recent publication represents the time stamp of a link. Link weights were computed using method introduced in [15] which is explained here. Let i and j are two authors and δ_i^k and δ_j^k are indicator functions. If author i is an author of paper k then $\delta_i^k = 1$ and zero otherwise. If paper k has n_k authors, the



(a) Astro-ph (b) Condmat-ph

(c) Hep-ph (d) Hep-th

Fig. 3 Variation of F-measure with decaying factor α (coauthorship data).

weight of collaboration w_{ij} between two authors i and j is computed as the summation of all coauthored papers;

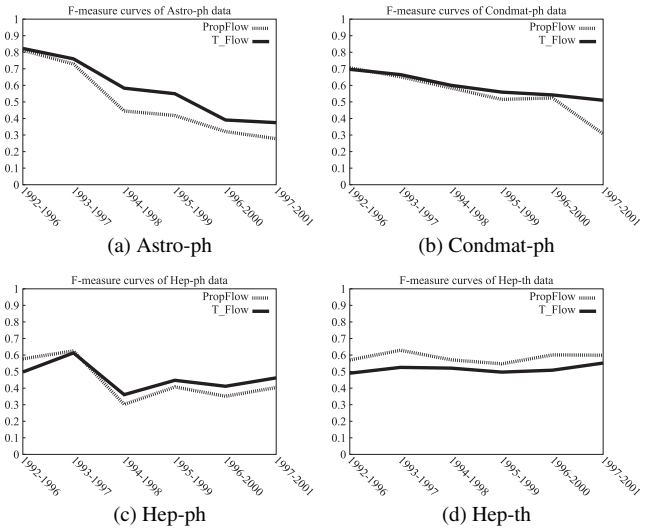
$$w_{ij} = \sum_k \frac{\delta_i^k \delta_j^k}{n_k - 1} \tag{16}$$

We train the decision tree algorithm using five consecutive years of coauthor data to predict links in the following year. For example, data from 1992 to 1996 is used as training data to predict links emerged in the year 1997. The unit of time for the coauthorship data is years.

Link prediction performance of *T_Flow combination* with the variation of α for each coauthorship data is shown in Fig. 3. We obtained the highest average F-measures at different α values for different subject areas. The activeness of links in coauthorship networks are not change rapidly as authors work together for long time to publish research papers. Therefore, the influence of coauthorship on link activeness is proportionately high. The other notable characteristic is that Astro-ph, Condmat-ph and Hep-ph coauthorship networks have high clustering coefficients and mean degrees as shown in Table 4. Higher clustering coefficients and mean degrees in the recent years tells that authors tends to interact (via publications) with more coauthors as networks grow with the time. More interactions makes networks more active and *T_Flow* combination perform better than *PropFlow* combination. In contrast, *PropFlow* combination performs significantly better than *T_Flow* combination for Hep-th data as shown in Fig. 3 (d). As shown in Table 4, Hep-th coauthorship network has low clustering coefficients and low mean degrees. This observation tells that this network is less active compared to the other subject areas. In other words, the authors rarely make new coauthorships. This phenomenon could specific to the network. In our experiments, we have assumed that the average time taken for a publication is one year. However, it takes more than one year in some research areas to make a publication. In such kind of situations, we have to choose the time unit depend-

Table 5 Comparison of *PropFlow* combination and *T_Flow* combination for coauthorship data.

Data set (Subject area)	Feature combination	Avg. Precision	Avg. Recall	Avg. F-measure
Astro-ph	*PropFlow*	0.7003	0.4208	0.5005
	T_Flow	**0.7394**	**0.5005**	**0.5802**
Condmat-ph	*PropFlow*	0.6573	0.4872	0.5480
	T_Flow	**0.7095**	**0.5208**	**0.5960**
Hep-ph	*PropFlow*	0.6795	0.3438	0.4443
	T_Flow	**0.6963**	**0.3525**	**0.4654**
Hep-th	*PropFlow*	0.6775	**0.5180**	**0.5862**
	T_Flow	**0.7381**	0.3973	0.5157



(a) Astro-ph (b) Condmat-ph

(c) Hep-ph (d) Hep-th

Fig. 4 Variation of F-measure with network growth (coauthorship data).

ing on the interaction time.

Table 5 shows summary of comparison of *PropFlow combination* and *T_Flow combination* for coauthorship data. We tested *T_Flow combination* using two-loop cross validation method for determining α and 10-fold cross validation for computing the results. The results shows that average F-measure of *T_Flow combination* is better than the average F-measure of *PropFlow combination*. In fact, *T_Flow* combination shows significant improvement in average F-measure for Astro-ph data. The results implies that the information flow via active links is a vital factor for link prediction.

In our further analysis, we observed that the difference between F-measure values of *PropFlow* and *T_Flow* combinations increase for recent coauthorship networks as shown in Figs. 4 (a), (b), and (c). In other words, *T_Flow* combination shows better performances on recent data sets which has higher clustering coefficient and mean degrees. Further, we obtained the highest F-measure for Condmat-ph data in the experimental results shown in Table 5. This means that the decay of information flow per unit time in Condmat-ph data is higher than the other subject areas. Such kind of data are appropriate to study the correlation between dynamic behavior of networks (network growth) and performance of *T_Flow* algorithm.

Table 6 Statistics of Condmat-ph data.

Training data	Nodes	Edges	Clustering coefficient	Mean degree
D1 (1997–2001)	40786	278152	0.7336	6.81
D2 (1998–2002)	46124	328432	0.7348	7.11
D3 (1999–2003)	50632	373934	0.7347	7.38
D4 (2000–2004)	55425	424116	0.7349	7.65
D5 (2001–2005)	59742	467608	0.7357	7.82
D6 (2002–2006)	62802	493634	0.7367	7.86

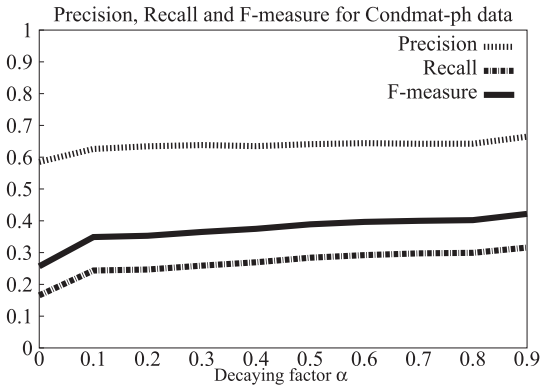


Fig. 5 Performance of *T_Flow combination* for different α values (Condmat-ph).

Table 7 Comparison of *PropFlow combination* and *T_Flow combination* for Condmat-ph data.

Feature combination	Avg. Precision	Avg. Recall	Avg. F-measure
PropFlow	0.5852	0.1655	0.2567
T_Flow	**0.6637**	**0.3258**	**0.4302**

4.3 Experiment with Condmat-ph Data

We carried out further experiments to investigate the performance of *T_Flow* algorithm when networks change rapidly. More recent network data shows rapid changes. Hence, we used six different network data sets extracted from Condmat-ph publications from 1997 to 2007. Statistics of the data sets are shown in Table 6 and experimental settings are the same as in Sect. 4.2. Clustering coefficients and mean degrees are almost same for six data sets. Link prediction performance of *T_Flow combination* with the variation of α is shown in Fig. 5. Comparison of *PropFlow* combination with *T_Flow* combination is shown in Table 7. We tested *T_Flow combination* using two-loop cross validation method for determining α and 10-fold cross validation for computing the results. The results shows a significant improvement for *T_Flow* combination. It implies that *T_Flow* algorithm is more sensitive for rapid changes in link activeness and hence, shows better performance for dynamic networks.

5. Conclusion

In this paper, we introduced a new algorithm called *T_Flow*

based on information flow which can be used for link prediction in social networks. *T_Flow* algorithm computes *T_Flow* using activeness of links and link weights. The main characteristic of *T_Flow* algorithm is that it considers the impact of link activeness for information flow which has not been discussed in the previous method. We combined the activeness of links and link weights in *T_Flow* algorithm and investigated how it affect the information flow which is a vital factor for link prediction. The experimental results shows that *T_Flow* algorithm outperform the previous *PropFlow* algorithm which only considers the impact of link weights for information flow. Thus, *T_Flow* algorithm is better for link prediction in social networks where the link activeness varies over time.

Acknowledgments

We would like to be thankful to the editor and reviewers for their valuable comments and sharing knowledge with us.

References

[1] L.A. Adamic and E. Adar, "Friends and neighbors on the web," Social Networks, vol.25, pp.211–230, 2003.
[2] L. Backstrom and J. Leskovec, "Supervised random walks: Predicting and recommending links in social networks," Proc. Forth International Conference on Web Search and Web Data Mining, pp.635–644, 2011.
[3] N.V. Chawla, K.W. Bowyer, L.O. Hall, and W.P. Kegelmeyer, "Smote: Synthetic minority over-sampling technique," J. Artificial Intelligence Research, vol.16, pp.321–357, 2002.
[4] M.A. Hasan, V. Chaoji, S. Salem, and M. Zaki, "Link prediction using supervised learning," Proc. SDM 06 Workshop on Link Analysis, Counterterrorism and Security, 2006.
[5] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I.H. Witten, "The weka data mining software: An update," SIGKDD Explor. Newsl., vol.11, no.1, pp.10–18, 2009.
[6] I. Kahanda and J. Neville, "Using transactional information to predict link strength in online social networks," Proc. Third International Conference on Weblogs and Social Media, pp.74–81, 2009.
[7] H. Kashima and N. Abe, "A parameterized probabilistic model of network evolution for supervised link prediction," Proc. 6th International Conference on Data Mining, pp.340–349, 2006.
[8] H. Kashima, T. Kato, Y. Yamanishi, M. Sugiyama, and K. Tsuda, "Link propagation: A fast semi-supervised learning algorithm for link prediction," Proc. SIAM International Conference on Data Mining, pp.1099–1110, 2009.
[9] D. Liben-Nowell and J. Kleinberg, "The link prediction problem for social networks," Proc. 12th International Conference on Information and Knowledge Management, pp.556–559, 2003.
[10] R.N. Lichtenwalter, J.T. Lussier, and N.V. Chawla, "New perspectives and methods in link prediction," Proc. 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp.243–252, 2010.
[11] C.D. Manning, P. Raghavan, and H. Schütze, Introduction to Information Retrieval, Cambridge University Press, 2008.
[12] L. Munasinghe and R. Ichise, "Time score: A new feature for link prediction in social networks," IEICE Trans. Inf. & Syst., vol.E95-D, no.3, pp.821–828, March 2012.
[13] L. Munasinghe and R. Ichise, "Exploiting information flow and active links for link prediction in social networks," Proc. 26th Annual Conference of the Japanese Society for Artificial Intelligence, 2012.
[14] M.E.J. Newman, "Clustering and preferential attachment in growing networks," Phys. Rev. E, vol.64, no.2, 025102, July 2001.
[15] M.E.J. Newman, "Scientific collaboration networks. ii. shortest paths, weighted networks, and centrality," Phys. Rev. E, vol.64, no.1, 016132, June 2001.
[16] M. Pavlov and R. Ichise, "Finding experts by link prediction in co-authorship networks," Proc. Workshop on Finding Experts on the Web with Semantics, pp.42–55, Nov. 2007.
[17] J.R. Quinlan, C4.5: Programs for machine learning, Morgan Kaufmann, 1993.
[18] M. Sachan and R. Ichise, "Using abstract information and community alignment information for link prediction," International J. Engineering and Technology, vol.2, no.4, pp.334–339, 2010.
[19] J. Scripps, P.-N. Tan, F. Chen, and A.-H. Esfahanian, "A matrix alignment approach for link prediction," Proc. 19th International Conference on Pattern Mining, pp.1–4, 2008.
[20] B. Viswanath, A. Mislove, M. Cha, and K.P. Gummadi, "On the evolution of user interaction in facebook," Proc. 2nd ACM SIGCOMM Workshop on Social Networks, Aug. 2009.
[21] C. Wang, V. Satuluri, and S. Parthasarathy, "Local probabilistic models for link prediction," Proc. 7th IEEE International Conference on Data Mining, pp.322–331, 2007.
[22] T. Wohlfarth and R. Ichise, "Semantic and event-based approach for link prediction," Proc. 7th International Conference on Practical Aspects of Knowledge Management, pp.50–61, 2008.

Lankeshwara Munasinghe received his B.Sc. degree in Statistics and Computing from University of Kelaniya, Srilanka in 2002. He is currently a Ph.D. candidate at the Graduate University for Advanced Studies in Japan. His research interests include machine learning and data mining.

Ryutaro Ichise received his Ph.D. degree in computer science from Tokyo Institute of Technology, Tokyo, Japan, in 2000. From 2001 to 2002, he was a visiting scholar at Stanford University. He is currently an associate professor in the Principles of Informatics Research Division at the National Institute of Informatics in Japan. His research interests include machine learning, semantic web, and data mining.