A Table of Notations

Table 4: Key notations used in this paper.

Symbol	*Definition*
$G(V, E)$	A social network graph
n	number of vertices in G
m	number of edges in G
$G_i(V_i, E_i)$	i^{th} component (subnetwork) in $G(V, E)$
Γ	A set of subnetworks (components)
m'	$\max_{E_i \subseteq E} \lvert E_i \rvert$
k	number of seeds to be selected
ℓ	number of subnetworks
R	number of rounds of simulation
β^i	A COG-sublist
Υ	A set of COG-sublists
$\mathcal{M}$	MAG-list
S	seed set
S_i	seed nodes selected from $G_i(V_i, E_i)$
$\Omega(\cdot)$	conformity index
$\Phi(\cdot)$	influence index
$\overrightarrow{uv}$	the edge pointing from u to v
$\sigma_i(\cdot)$	influence function under cascade model C_i
T	number of iterations in gain computation
$\mathbb{T}$	A topic
$E_{\mathbb{T}}$	edge correlated with topic $\mathbb{T}$
$G_{\mathbb{T}}$	subgraph correlated with topic $\mathbb{T}$
$\Omega_{\mathbb{T}}(\cdot)$	conformity index with respect to topic $\mathbb{T}$
$\Phi_{\mathbb{T}}(\cdot)$	influence index with respect to topic $\mathbb{T}$

B Example related to Influenceability-based Approach

Consider the network in Fig. 15 where each node denotes an individual. Suppose we aim to select a single seed node ($k = 1$) to propagate a piece of information (*e.g., iPad*). Suppose we use the influenceability-based technique [16] to determine the seed. Recall that "influenceability" is defined as the ratio of propagated actions divided by the total number of actions. For example, in Fig 15 user v_1 performed 4 actions on topic A_1, A_2, A_3 and A_4, respectively. Among these actions, actions on A_2, A_3 and A_4 are propagated from v_5. Therefore, the influenceability of v_1 is $3/4 = 0.75$. Similarly, the influenceability of other nodes can be computed and are shown in Table 5 along with their influence and conformity indices. We can computed the expected influence for each node by using the influenceability as the propagation probability. The results are shown in Table 6 along with related approaches including our conformity-aware models. Hence, we may select v_5 as the seed. However, v_5 exhibits very small influence whereas at the same time v_1 exhibits low conformity. In other words, v_1 is not easily influenced by v_5 and as a result $\overrightarrow{v_5v_1}$ is hardly activated during influence propagation. Hence this is not the best choice when conformity of nodes are taken into account.

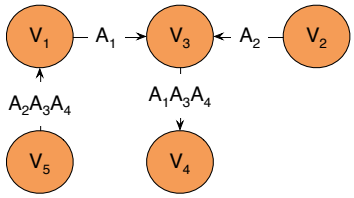


Fig. 15: A network.

Table 5: Nodes' influence and conformity indices.

Node ID	$\Phi(\cdot)$	$\Omega(\cdot)$	$\Phi_1(\cdot)$	$\Omega_1(\cdot)$	*influenceability*
v_1	0.68	0.21	0.70	0.17	0.75
v_2	0.68	0.11	-	-	0
v_3	0.18	0.94	0.70	0.70	0.5
v_4	0.03	0.21	0.17	0.70	1
v_5	0.18	0.11	-	-	0

Table 6: Expected influence size of nodes in Fig. 15.

Model	$\sigma(v_1)$	$\sigma(v_2)$	$\sigma(v_3)$	$\sigma(v_4)$	$\sigma(v_5)$
IC	1.75	1.75	1.5	1	1.875
WC	1.67	1.67	2	1	1.83
[16]	2	2	2	1	2.5
C^2	1.66	1.66	1.04	1	1.06
C^3 (for A_1)	1.73	1	1.49	1	1

C Proof of Theorem 1

According to Definition 1, for each vertex u its influence index $\Phi(u)$ can be computed as the following.

$$\Phi(u) = \sum_{\overrightarrow{u'u} \in E^+} \Omega(u') - \sum_{\overrightarrow{u''u} \in E^-} \Omega(u'')$$

If we denote $\mathbb{I} = (\Phi(u_1), \Phi(u_2), \ldots, \Phi(u_\ell))^\top$ and $\mathbb{C} = (\Omega(u_1), \Omega(u_2), \ldots, \Omega(u_\ell))^\top$ for $V = \{u_1, u_2, \ldots, u_\ell\}$, then the computation of both indices in each iteration can be represented as:

$$\begin{cases} \mathbb{I} = \mathbb{A}_+^\top \mathbb{C} - \mathbb{A}_-^\top \mathbb{C} \\ \mathbb{C} = \mathbb{A}_+ \mathbb{I} - \mathbb{A}_- \mathbb{I} \end{cases}$$

If we substitute $\mathbb{C}$ in the first equation using the second equation, then the first line turns into the following:

$$\begin{aligned} \mathbb{I}_{k+1} &= \frac{1}{Z}(\mathbb{A}_+^\top - \mathbb{A}_-^\top)(\mathbb{A}_+ - \mathbb{A}_-)\mathbb{I}_k \\ &= \frac{1}{Z}(\mathbb{A}_+ - \mathbb{A}_-)^\top(\mathbb{A}_+ - \mathbb{A}_-)\mathbb{I}_k \end{aligned}$$

where Z is a normalizing factor such that $\|\mathbb{I}_{k+1}\| = 1$. If we compute $\mathbb{I}_{k+1}$ using $\mathbb{I}_k$ for $k = 1, 2, \ldots, n$ recursively, then $\mathbb{I}_{n+1}$ should be the unit vector along the direction of

$$((\mathbb{A}_+ - \mathbb{A}_-)^\top(\mathbb{A}_+ - \mathbb{A}_-))^n(\mathbb{A}_+ - \mathbb{A}_-)^\top(1, 1, \ldots, 1)^\top.$$

Similarly, $\mathbb{C}_{n+1}$ should be the unit vector along the direction of

$$((\mathbb{A}_+ - \mathbb{A}_-)(\mathbb{A}_+ - \mathbb{A}_-)^\top)^{n+1}(1, 1, \ldots, 1)^\top.$$

If M is a symmetric matrix, and v is a vector not orthogonal to the principal eigenvector $\omega_1(M)$, then the unit vector in the direction of $M^k v$ converges to $\omega_1(M)$ as k increases.

Comparing with our case, $(1, 1, \ldots, 1)^\top$ is not orthogonal to $\omega_1((\mathbb{A}_+ - \mathbb{A}_-)(\mathbb{A}_+ - \mathbb{A}_-)^\top)$, thus $\mathbb{C}_k$ converges. Similarly, $\mathbb{I}_k$ also converges.

In summary, both $\Phi(u)$ and $\Omega(u)$ converge.

D Initial Value of CASINO

Besides setting the initial values of indices for each node to 1, we test two other initial value settings. One is to set the initial value for each node by sampling from a uniform distribution ranging from 0 to 1 (denoted as *UniDist* for brevity); the other method is to set the initial values proportional to the degree of the nodes where both indices for the nodes with the largest degree are set to 1 (referred to as *DegreePro* for brevity). We use datasets in Tables 8 and 9 to investigate the impact of these initial values. We observe that all approaches converge to the same result set but with different number of iterations. The number of iterations for different initial values for each dataset is reported in Table 7. Since the number of iterations does not differ significantly among different strategies, we set all initial values to 1 for simplicity.

Table 7: Number of Iterations before Convergence

Initial Value	Hep	Phy	Twitter	Wiki	LJ
All set to 1	33	31	30	35	41
UniDist	38	32	34	37	42
DegreePro	35	29	31	34	40

E Cascade Models

Majority of the literature on influence maximization have focused on the following cascade models as defined in [23].

- *Independent cascade (IC) model.* Let A_i be the set of nodes that are influenced in the i-th round and $A_o = |S|$. For any $(u, v) \in E$ such that u is already in A_i and v is not yet influenced, v is influenced by u in the next $(i+1)$-th round with an independent probability p, which is referred to as the *propagation probability*. Thus, if there are t neighbors of v that are in A_i, then $v \in A_{i+1}$ with probability $1 - (1 - p)^t$. This process is repeated until A_{i+1} is empty.
- *Weighted cascade (WC) model.* The WC model can be considered as an instance of IC model [23]. Let $(u, v) \in E$. In this model, if u is influenced in round i, then v is influenced by u in round $(i + 1)$ with probability $1/v.degree$. Thus, if v has t neighbors influenced at the i-th round then the probability for a node v to be influenced in the next round is $1 - (1 - 1/v.degree)^t$.
- *Linear threshold (LT) model.* Here each node v has a threshold θ_v uniformly and randomly chosen from 0 to 1; this represents the weighted fraction of v's neighbors that must become influenced (active) in order for v to be influenced. All nodes that were influenced in step $(i - 1)$ remains so in step i, and any node v is influenced when the total weight of its influenced neighbors is at least θ_v.

F Proof of Theorem 2

Let S_1 and S_2 be two sets of nodes such that $S_1 \subseteq S_2$. $R(v, X)$ denotes the set of all nodes that can be reached from v on all the activated edges that are in X. Consider the expression of $\sigma^X(S_1 \cup \{v\}) - \sigma^X(S_1)$. It denotes the number of elements in $R(v, X)$ that are not already in $\bigcup_{u \in S_1} R(u, X)$, which is at least as large as the number of elements in $R(v, X)$ that are not in $\bigcup_{u \in S_2} R(u, X)$. That is $\sigma^X(S_1 \cup \{v\}) - \sigma^X(S_1) \geq \sigma^X(S_2 \cup \{v\}) - \sigma^X(S_2)$, which means that the function $\sigma^X(\cdot)$ is submodular. Moreover, we have shown that $\sigma(\cdot)$ can be computed from $\sigma^X(\cdot)$ using Equation (1). It means $\sigma(\cdot)$ is a non-negative linear combination of another submodular function $\sigma^X(\cdot)$. Hence $\sigma(\cdot)$ is also submodular.

G Proof of Theorem 3

According to Definition 9, $\sigma(S)$ can be represented as the following.

$$\sigma(S) = \max_{\sum |S_i| = k} \sum \sigma_i(S_i)$$

Assume $S' \subset S, v \in V_t \setminus S_t$ where $t \in \{1 \ldots \ell\}$ and $S = S_1 \cup S_2 \cup \ldots \cup S_\ell, S' = S'_1 \cup S'_2 \cup \ldots \cup S'_\ell$, then $S'_i \subseteq S_i$. Besides, the following expression holds as $S_i \cap S_j = \emptyset \; \forall \; 0 < (i, j) \leq \ell$.

$$\begin{aligned} \sigma(S \cup \{v\}) - \sigma(S) = &\quad \sigma_t(S_t \cup \{v\}) - \sigma_t(S_t) \\ \sigma(S' \cup \{v\}) - \sigma(S') = &\quad \sigma_t(S'_t \cup \{v\}) - \sigma_t(S'_t) \end{aligned}$$

As $S'_t \subseteq S'$ and the influence function $\sigma_t(\cdot)$ is submodular, then $\sigma_t(S_t \cup \{v\}) - \sigma_t(S_t) \leq \sigma_t(S'_t \cup \{v\}) - \sigma_t(S'_t)$ holds according to the definition of submodularity. Thus, $\sigma(S \cup \{v\}) - \sigma(S) \leq \sigma(S' \cup \{v\}) - \sigma(S')$ holds too, which means that the influence function $\sigma(S)$ is submodular.

H Proof of Theorem 4

Without loss of generality, let Δ be the set of edges that are cut during the partitioning phase. Let v_i^s and v_i^e be the source node and end node of a cut edge $m_i (m_i \in \Delta)$, respectively. Assume that the expected influence of v_i^s after cutting m_i is $\sigma'(v_i^s)$. Then before the cut the influence of v_i^s can be calculated as $(1 - p_{m_i})\sigma'(v_i^s) + p_{m_i}(\sigma'(v_i^s) + \sigma(v_i^e)) = \sigma'(v_i^s) + p_{m_i}\sigma(v_i^e)$, where p_{m_i} represents the influence probability of edge m_i, which depends on the cascade model. Hence, the cut edge m_i will cause v_i^s's expected influence reduced by $p_{m_i}\sigma(v_i^e)$. Generally, for $\overrightarrow{v_i^{-n} v_i^{-n+1}}, \ldots, \overrightarrow{v_i^{-1} v_i^s} \in E$, the removal of edge m_i (edge $\overrightarrow{v_i^s v_i^e}$) will result in a deduction of $\sigma(v_i^{-n})$ by $p_{m_i}\sigma(v_i^e) \prod_{j=1}^{n} p_{m_i}^{-j}$, where $p_{m_i}^{-j}$ represents the influence probability of edge $\overrightarrow{v_i^{-j} v_i^{-j+1}}$. Specially, $v_i^0 = v_i^s$.

Obviously, the influence probability $p_{m_i} \leq 1$ for $\forall m_i \in \Delta$. Therefore, the loss of expected influence for an arbitrary node, denoted by $\chi(V)$ (*i.e.*, $\chi(V) = \max_{v \in V}(\sigma^0(v) - \sigma(v))$) is upper bounded by $\sum_{m_i \in \Delta} p_{m_i}\sigma(v_i^e)$ when all edges in Δ share the same source node. In contrary, the maximal loss of expected influence for an arbitrary node is lower bounded by $\max_{m_i \in \Delta} p_{m_i}\sigma(v_i^e)$ when none of the edges in Δ share the same source node.

Table 8: Description of real-world networks.

network	*nodes*	*edges*	*components*	m'
Phy	37,154	231,584	3,883	134,358
Hep	15,233	58,891	1,781	19,630
Wiki-talk	2,394,385	5,021,410	34	5,018,445
LiveJournal	4,847,571	68,993,773	1,145,331	65,825,429

Table 9: Description of the context-aware *Twitter* network.

#tweets	*#trends*	*#tweeters*	*#edges*	*#components*	m'
1,054,261	21,917	576,894	1,230,748	24	271,319

Table 10: Top 10 authors with the highest influence index and conformity index.

Rank	Influential twitter (#positive in-links/#in-links)			Conformer twitter (#positive out-links/#out-links)		
	All	*Top-1*	*Top-2*	*All*	*Top-1*	*Top-2*
1	142987924 (66/73)	3453454 (13/13)	950596 (31/34)	51389816 (35/37)	121836131 (14/16)	49276778 (101/144)
2	49276778 (61/82)	56068621 (11/11)	190108655 (11/11)	172039151 (31/34)	105332925 (13/14)	202346609 (45/61)
3	119394881 (60/77)	3984874 (10/10)	3498571 (8/9)	177173204 (30/35)	177255919 (11/12)	197538544 (26/30)
4	231134989 (55/71)	133282617 (11/11)	147327886 (5/5)	143062806 (27/34)	193206052 (11/12)	184930795 (22/26)
5	2109823 (56/72)	199855121 (7/7)	49126931 (5/5)	128118710 (25/33)	36525648 (9/10)	148335502 (21/23)
6	92503401 (55/78)	8234375 (5/5)	121158546 (5/5)	130414633 (30/41)	90723076 (7/8)	171387567 (17/20)
7	206661373 (51/66)	1465130 (3/3)	129009252 (5/5)	4782790 (23/30)	123606641 (6/8)	126407259 (18/22)
8	220490093 (46/60)	2894822 (3/3)	79897503 (4/4)	125551983 (22/34)	51513825 (6/6)	114455733 (14/20)
9	168175236 (40/51)	21755211 (2/2)	83629945 (4/4)	91930055 (21/28)	203774695 (5/6)	217826740 (15/20)
10	171287044 (41/62)	4051581 (2/2)	166830172 (4/4)	145339829 (22/31)	203780314 (4/4)	159724683 (12/17)

I Proof of Theorem 5 (Sketch)

The time complexity of the indices computation step using CASINO (Line 2 in Algorithm 4) is $O(k'm'n')$ where k' is the number of iterations in influence and conformity indices computation. The time complexity of the influence maximization step (Lines 4-6 in Algorithm 4) is $O(kTRm')$. Hence, the time complexity of CINEMA is $O(k'm'n' + kTRm')$.

J Pseudocode for Reduce Phase

The pseudocode of the Reduce phase is outlined in Algorithm 8.

Algorithm 8: Pseudo-code of Reduce phase.

Input: Intermediate key/value pairs for subgraph G_i: $((u, \sigma_i(u)), (i, j))$ and a group of these pairs for subgraphs $G_{i'}(i' \neq i)$: $((u', \sigma_{i'}(u')), (i', j'))$
Output: The key/value pairs for each $u \in G_i$: $((u, \sigma_i(u)), r)$

```
1 begin
2     foreach i' ≠ i do
3         if σ_i'(u') ≥ σ_i(u) then
4             r = r + 1
5     foreach ((u, σ_i(u)), (i, j)) do
6         return ((u, σ_i(u)), j = j + r − 1)
```

K Statistics of Datasets

The statistics of the five networks used in experimental study are given in Tables 8 and 9.

L Additional Experimental Results

L.1 A Case Study in Twitter

Table 10 shows IDs of top-10 authors who exhibit the highest influence index and conformity index for the top-2 topics as well as for all topics. Consider the top two twitters for all topics. The author `'142987924'` who has the highest influence index receives 66 conforming edges out of 73 in-links over 22 topics. Similarly, the author `'49276778'` receives 61 conforming edges out of 82 in-links over 24 topics. On the other hand, the author `'51389816'` who exhibits the highest conformity index initiates 35 conforming edges out of 37 out-links over 37 topics indicating that she has high chance to conform to others' opinions in almost all the topics she is involved in. Furthermore, we can make the following observations. Firstly, none of the top-10 authors occupies a position in *both* indices for each category (*all*, *top-1*, and *top-2*). Secondly, the top-10 individuals having highest influence and conformity indices are different for different topics. This confirms our hypothesis that social influence phenomenon is context-sensitive as same individual may exhibit different influence and conformity over different topics of social interactions.

We also adopted a regression function of SVM^{light} and use both influence and conformity indices as features, where presence of positive edges are labeled as 1 and negative edges are

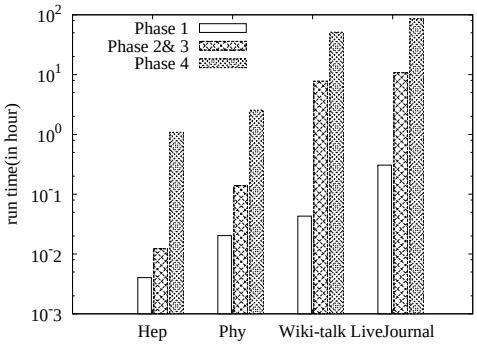


Fig. 16: Cost of Phases 1–4.

labeled as -1. In the experiment we adopt the classifier ICAPN which takes into account the topic information associated with each edge. That is, the following features for each edge $\overrightarrow{uAv}$ are used to train the model: $d^+_{in}(v)$ (positive in-degree), $d^-_{in}(v)$, $d^+_{out}(u)$, $d^-_{out}(u)$, $d_{in}(v)$, $d_{out}(u)$, $C(u,v)$ (number of common neighbors), $\Phi(v)$, $\Omega(u)$, $\Phi_A(v)$ and $\Omega_A(u)$. Figure 18 plots the prediction accuracies of the relevant classifiers. Observe that in both figures ICAPN outperforms the rest baseline methods. All these evidences demonstrate that by leveraging on the influence and conformity indices in topic-based subgraphs, the proposed model leads to superior prediction performance for both positive edge presence and edge sign prediction tasks.

L.2 Cost of Phases 1-4

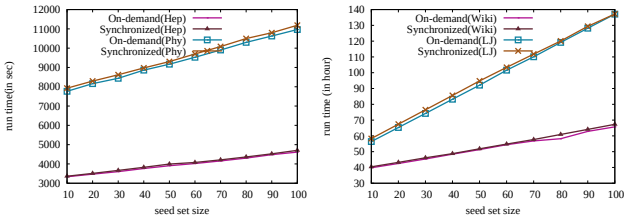
Next, we analyze the cost of Phases 1–4 of CINEMA. Fig. 16 compares the running times of these phases for the four datasets. Since the running time of MAG-list construction is significantly smaller than the rest, we plot the total running time of Phases 1 and 2. Observe that the seed selection phase dominates the running time agreeing with our analysis in Sec. 5.2. Note that in order to ensure fair comparison with *Hep* and *Phy*, for *Wiki-talk* and *LiveJournal* we depict only the partitioning times of the ℓ-way partitioning algorithm and not its initial failed attempt to partition using BFS technique.

L.3 On-demand vs. Synchronized Update

Next, we compare the on-demand and synchronized update strategies introduced earlier and justify our choice of the former. Note that the choice of using one of these strategy only affects the update performance of MAG-list and COG-sublist and not the seed set quality. Figures 17a and 17b plot the comparison of the running times between the two strategies for different values of k. The running times of both strategies increase linearly with k. Besides, the on-demand strategy is slightly better than the synchronized one which also agrees with our discussion in the preceding section.

L.4 Comparative Study of Seeds Selection in Twitter

Table 11 reports *Twitter IDs* of seeds selected by different models for $k = 10$. The first four columns report the seeds

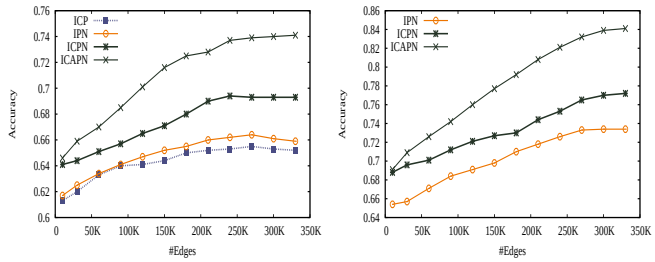


(a) *Hep* & *Phy* (b) *Wiki-talk* & *LiveJournal*

Fig. 17: [Best viewed in color] Update strategies

Table 12: Features involved in different approaches. (P: positive, N: negative, I: influence, C: conformity, A: topic)

Features \ Approaches	P	PN	IPN	ICP	ICPN	ICAPN
$d^+_{in}(v)$	✓	✓	✓	✓	✓	✓
$d^-_{in}(v)$	-	✓	✓	-	✓	✓
$d^+_{out}(u)$	✓	✓	✓	✓	✓	✓
$d^-_{out}(u)$	-	✓	✓	-	✓	✓
$d_{in}(v)$	-	✓	✓	-	✓	✓
$d_{out}(u)$	-	✓	✓	-	✓	✓
$C(u,v)$	✓	✓	✓	✓	✓	✓
$\Phi(u)$	-	-	✓	-	-	-
$\Phi(v)$	-	-	✓	✓	✓	✓
$\Omega(u)$	-	-	-	✓	✓	✓
context A	-	-	-	-	-	✓
$\Phi_A(v)$	-	-	-	-	-	✓
$\Omega_A(u)$	-	-	-	-	-	✓



(a) Positive edge presence (b) Edge sign prediction

Fig. 18: Context-aware prediction accuracy.

computed by CINEMA-C^3 for four different topics. The seeds set computed by *MixGreedy*-IC and CINEMA-C^2 over the entire network are shown in the last two columns. We can make two key observations. Firstly, the seeds identified by CINEMA-C^3 are completely distinct from those selected by context-unaware techniques (*MixGreedy* and CINEMA-C^2). In other words, it further strengthen our conclusion that the result quality can improve significantly if topic information is incorporated in the IM problem. Secondly, the seeds generated by CINEMA-C^2 and *MixGreedy*-IC are significantly different (only 4 out of 10 seeds appear in both sets) highlighting the importance of conformity-awareness for IM problem.

Table 11: Seeds selected from Twitter.

CINEMA-C^3				*MixGreedy*-IC	CINEMA-C^2
Mumford & Sons	WeLoveTokioHotel	BornThisWayFriday	Mubarak		
3453454	4093419	950596	388397	8994366	8994366
56068621	191059547	190108655	4725921	6837510	950596
3984874	114799747	3498571	5549	143131074	6837510
133282617	22418179	147327886	25817119	12732578	4976883
199855121	90276810	49126931	4112233	969858	106789932
8234375	201647517	121158546	957238	20735827	179500
4051581	146896900	79897503	3238537	1327826	20735827
2894822	97497115	4051581	69290548	2494788	124440574
202658279	204479139	193820280	190736000	4740643	4740643
780597	206998557	83629945	1314262	1111124	940898

L.5 Edge Prediction

In this experiment, we summarize how the influence and conformity indices can facilitate link prediction. A more detailed exposition of this issue is given in [29]. On one hand, we conduct experiments to predict the presence of positive edges. Note that in order to test the effect of negative edges in predicting the presence of positive edges, we adopted a regression function of SVM^{light} where presence of positive edges are labeled as 1 and negative edges are labeled as -1. We investigate how each of the aforementioned classifiers perform in predicting the presence of positive edges. On the other hand, we undertake another group of experiments to predict the signs of edges. We ensure that the training set and test set both contain equal number of positive and negative edges. Note that in a binary classification, positive edges and negative edges belong to two different classes. Our goal is to predict the signs of edges which maybe either positive or negative. In both experiments, we adopt the classifier ICAPN (Table 12) which takes into account the topic information associated with each edge. That is, the following features for each edge $\overrightarrow{uAv}$ are used to train the model: $d^+_{in}(v)$, $d^-_{in}(v)$ (*i.e.,* positive and negative in-degree of v), $d^+_{out}(u)$, $d^-_{out}(u)$ (*i.e.,* positive and negative out-degree of v), $d_{in}(v)$, $d_{out}(u)$ (*i.e.,* in- and out-degree of v), $C(u,v)$ (*i.e.,* number of common neighbors of u,v), $\Phi(v)$, $\Omega(u)$, $\Phi_A(v)$ and $\Omega_A(u)$. Figure 18 plots the prediction accuracies of the relevant classifiers over *Twitter* dataset by varying the training set. Each classifier contains different features described in Table 12. Observe that in both figures ICAPN outperforms the rest. Note that the performances of ICAPN and ICPN are similar when the training set is very small. This is because there may not be enough training edges in each topic-based subgraphs G_A when the training set is very small. Consequently, not enough information is available to accurately compute $\Phi_A(v)$ and $\Omega_A(u)$. All these evidences demonstrate that by leveraging on the influence and conformity indices in topic-based subgraphs, our indices computation model leads to superior prediction performance for both positive edge presence and edge sign prediction tasks.