电子科技大学
University of Electronic Science and Technology of China

Clustering On dynamic networks

Zhongjing Yu

Data Mining Lab, Big Data Research Center, UESTC
Email：yuzhongjing@foxmail.com
http://staff.uestc.edu.cn/yuzhongjing

为什么选这个题目（网络\图流上的聚类）：

静态网络上的聚类很早就被广泛关注， 各种方法相应提出。然而，现实生活中也存在这大量的动态网络，即随着时间的变化网络的结构会因点和边的变化而变化，特别是近年来随着网络的快速发展。在学术界大约在2010年左右大家也开始关注动态网络聚类的研究.动态网络也大体分为两种[1]：缓慢的演化和快速的演化。这里偏向的是快速的演化（流形式）。

在动态网络的演化过程中，由于数据的量大，速度快，必须使得聚类算法更加高效。然而，这里不能用传统的方法直接对每个时刻直接一步到位的得到社团结构，因此采用的是一种基于微簇的聚类方法，即先将数据进行粗略的聚类，得到一些微簇，再将微簇进行进一步的聚类。该topic通过从很经典的FacetNet（2009）算法讲起，直到2014年左右的比较经典的网络流聚类算法。

[1] Evolutionary Network Analysis : A Survey. ACM Computing Surveys, 2014

前沿推荐：

在网络流方面的算法是比较多的，不仅可以从上述的数据挖掘的角度来看，还可以从机器学习的角度来看。机器学习方面通常包括根据矩阵和数理统计的方面做，但是由于矩阵的角度做的话有两个实际的局限：存储（矩阵的存储对机器的要求较高）和线性操作（矩阵的操作只能是线性的操作，非线性的操作可能要借助核等，但这对网络流来说，复杂度通常很高）。因此，本人更偏向与数理统计。近几年对于社团发现(不仅限于社团发现)比较流行的统计方面的方法是随机块模型（Stochastic Block Model）[1][2][3][4][5]。

注：这部分需要一定的统计或概率知识（Dirichlet 过程等）。

[1]. Community detection on Evolution Graphs. NIPS 2016
[2]. Stochastic Block Transition Models for Dynamic Networks. AISTATS 2015
[3]. Integrating Community and Role Detection in Information Networks. (Yizhou Sun group).
[4].Graph Clustering Block-model and model free results. NIPS 2016.
[5]. Community Detection on Evolving Graphs. NIPS 2016.

Outline

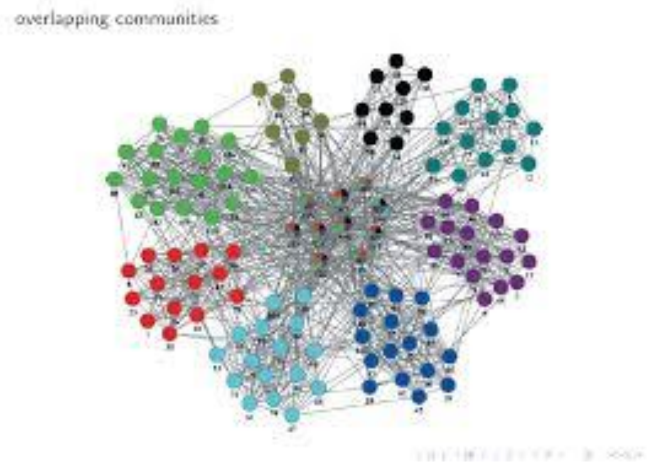
Background

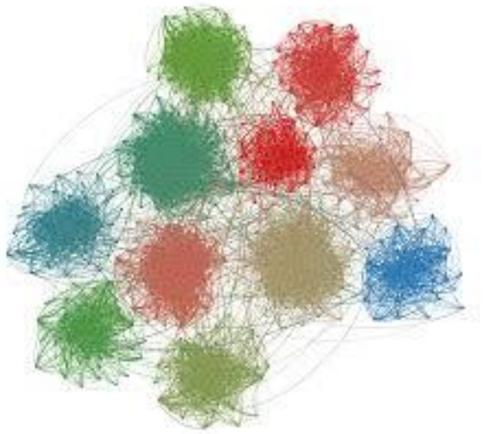
Clustering algorithms based on dynamic networks(graph stream)

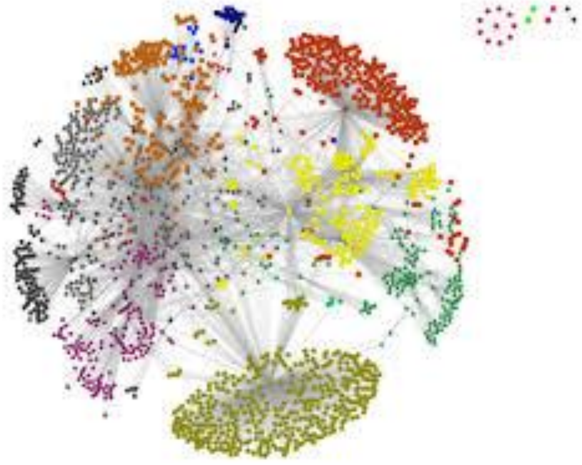
- FacetNet[Yu-RL et al. WWW’08]
- A particle-and-density ~~[Yuan et al. CIKM’13]
- Based on skeletal (Pei L et al. ICDE’14)
- **Local Weighted-Edge-based Pattern**(LWEP)[ICDM’13]

- Background

Clustering——static networks

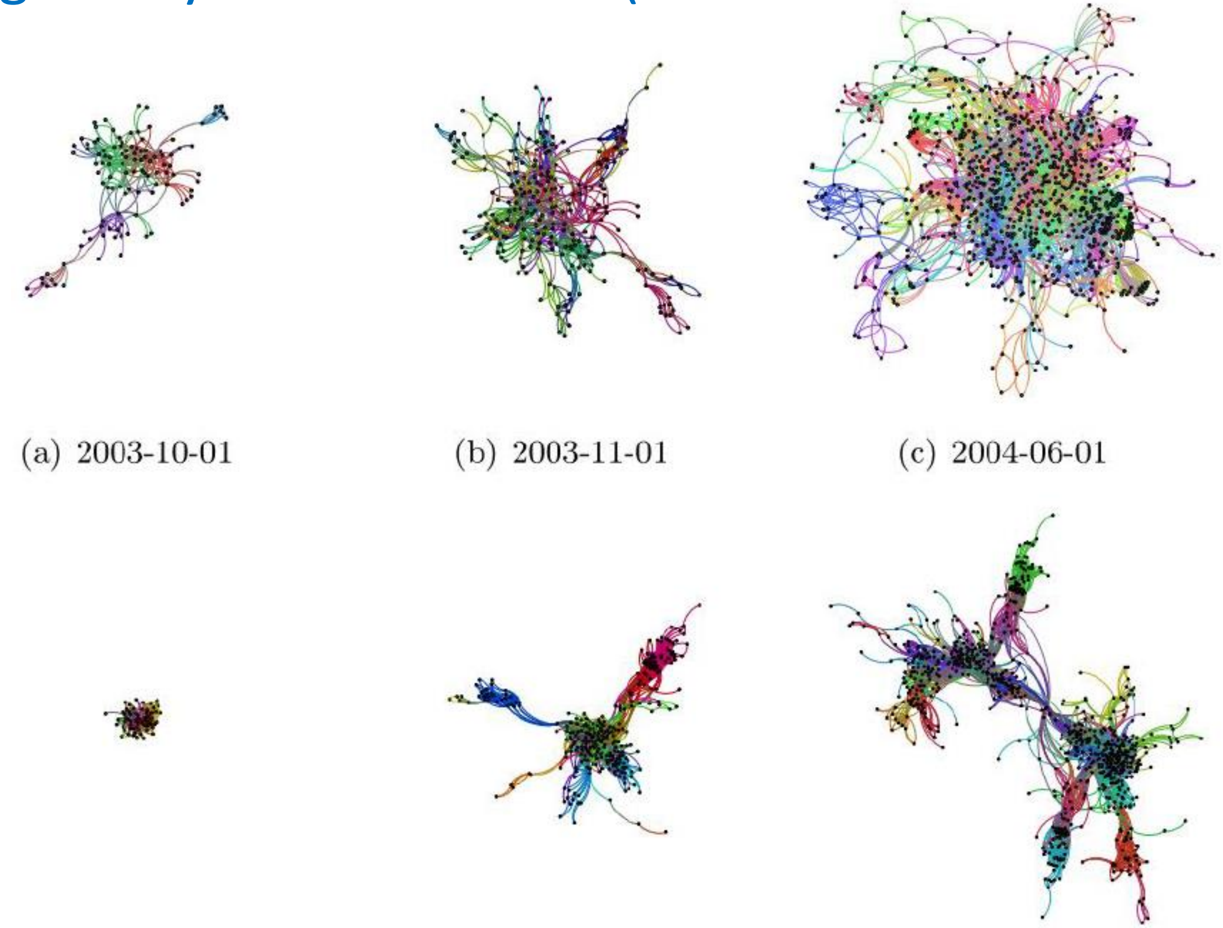






Background

Clustering——dynamic networks(slow evolution networks)



(a) 2003-10-01

(b) 2003-11-01

(c) 2004-06-01

AZUREUS

(d) 2001-02-01

(e) 2001-10-01

(f) 2003-01-01

JENA

Background

Clustering——dynamic networks(graph steam)

Initial Given a graph $G = (E, V)$

Time evolution New nodes, and new edges are added/deleted to the graph.

Notice : as time increasing, new edges may be same with older edges in different time. (Data set like DBLP).

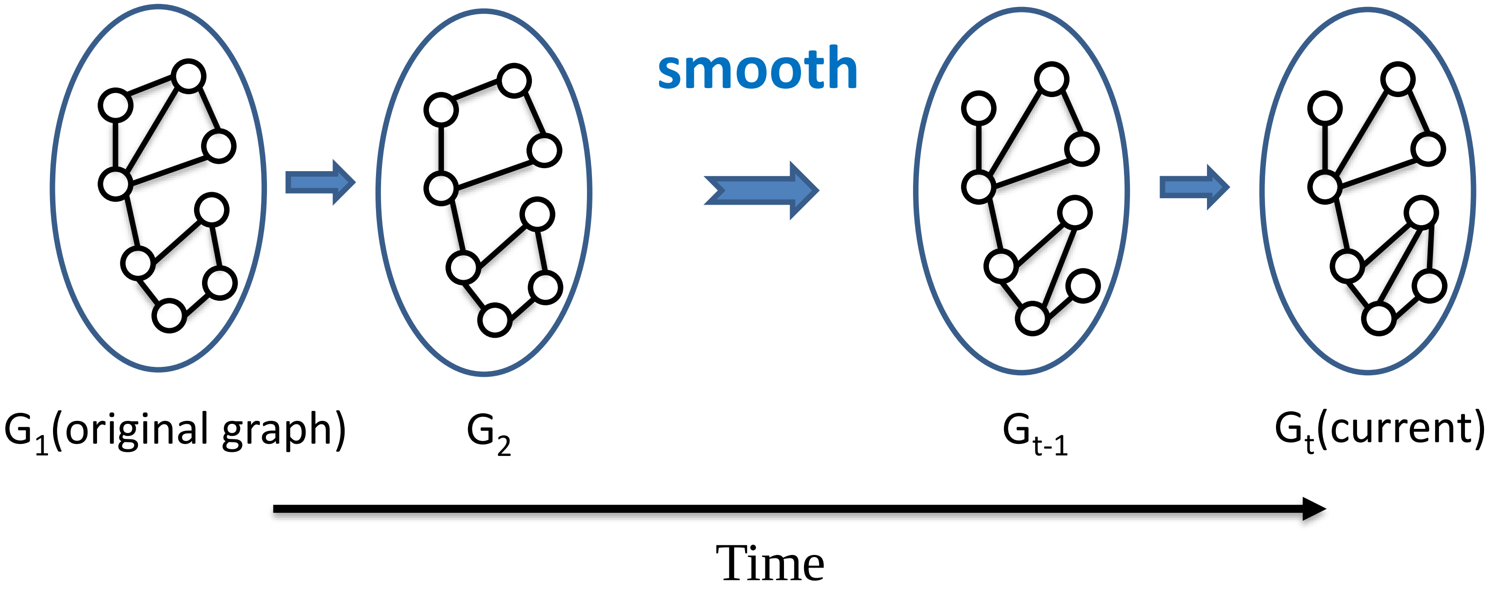
Clustering based on based on dynamic networks

Clustering based on dynamic networks

FacetNet

Basic Idea: communities not only **generate evolutions**, they also are regularized by the **temporal smoothness of evolutions**.

Method: it will discover communities that jointly maximize the **fit** to **the observed data** and **the temporal evolution** via ***NMF***.



Clustering based on dynamic networks

FacetNet(detail)

The **community structure** provides evidence about community evolutions and at the same time, the **evolutionary history offers hints** on what community structure is more appropriate.

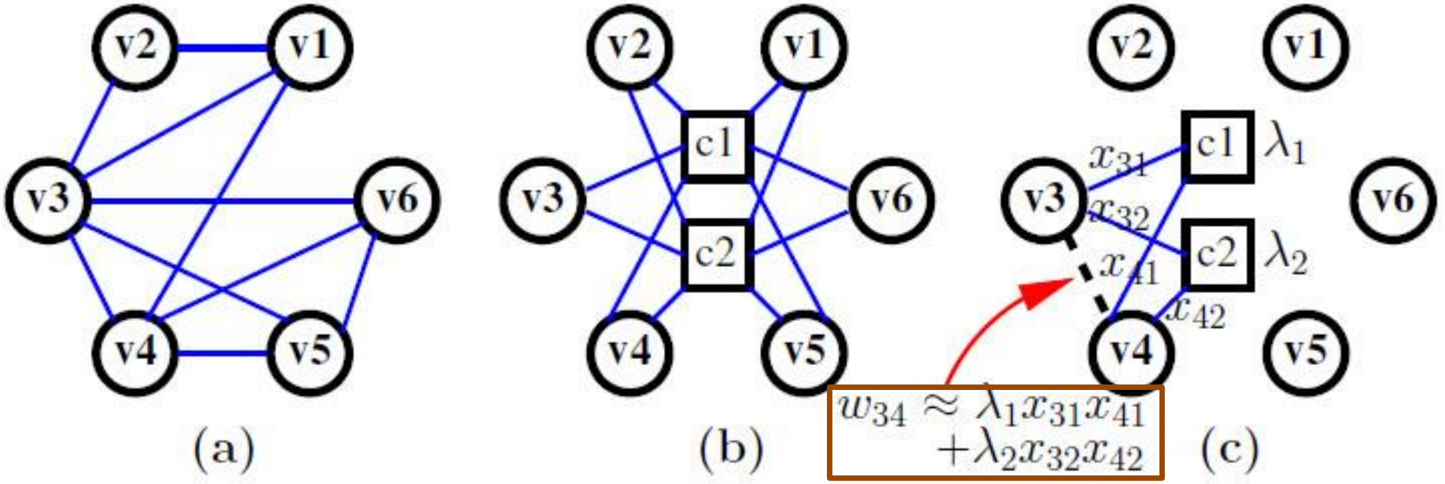
Construct cost function:

$$cost = \alpha \cdot \underbrace{\mathcal{CS}}_{\text{Snapshot Cost}} + (1-\alpha) \cdot \underbrace{\mathcal{CT}}_{\text{History Cost}}$$

Clustering based on dynamic networks

FacetNet(detail)

——Snapshot Cost



(a) (b) (c)

$w_{34} \approx \lambda_1 x_{31} x_{41} + \lambda_2 x_{32} x_{42}$

(Where c_1, c_2 is clusters. p_k is prior probability)

$$w_{ij} \approx \sum_{k=1}^{m} p_k \cdot p_{k\to i} \cdot p_{k\to j} \Longrightarrow W \approx X \Lambda X^T$$

$$\Downarrow$$

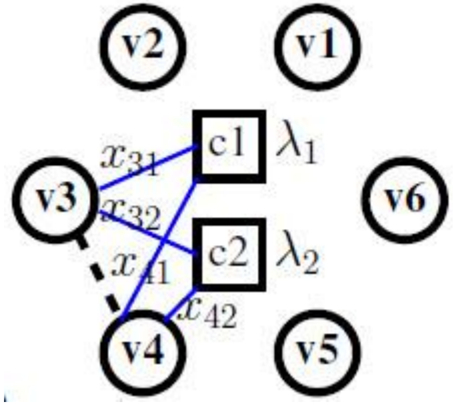
$$CS = D(W \| X \Lambda X^T)$$

Clustering based on dynamic networks

FacetNet(detail)

——*Temporal* Cost

Preventing unreasonably dramatic community evolution from time *t-1* to *t.*



The community structure is captured by $X\Lambda$

$Y \doteq X_{t-1}\Lambda_{t-1}$ (Y : community structure at t-1.)

⇩

$$CT = D(Y\|X\Lambda)$$

Assumption: the number of communities is fix.

Clustering based on dynamic networks

FacetNet(detail)

$$cost = \alpha \cdot D(W \| X \Lambda X^T) + (1-\alpha) \cdot D(Y \| X \Lambda)$$

$$x_{ik} \leftarrow x_{ik} \cdot 2\alpha \cdot \sum_j \frac{w_{ij} \cdot \lambda_k \cdot x_{jk}}{(X \Lambda X^T)_{ij}} + (1-\alpha) \cdot y_{ik}$$

$$\textit{then normalize such that } \sum_i x_{ik} = 1, \forall k$$

$$\lambda_k \leftarrow \lambda_k \cdot \alpha \cdot \sum_{ij} \frac{w_{ij} \cdot x_{ik} \cdot x_{jk}}{(X \Lambda X^T)_{ij}} + (1-\alpha) \cdot \sum_i y_{ik}$$

$$\textit{then normalize such that } \sum_k \lambda_k = 1.$$

Clustering based on dynamic networks

FacetNet(extensions)

Inserting and removing Nodes:

operation of matrix(remove or insert rows).

Changing community numbers:

deviation between the change for edges among communities

$$Q(\mathcal{P}_m) = \sum_{k=1}^{m} \left[\frac{\mathcal{A}(V_k, V_k)}{\mathcal{A}(V, V)} - \left(\frac{\mathcal{A}(V_k, V)}{\mathcal{A}(V, V)} \right)^2 \right]$$

$$\mathcal{A}(V_p, V_q) = \textstyle\sum_{i \in V_p, j \in V_q} w_{ij}$$

Soft Modularity Qs to get m(clusters)

Optimize : $Q_s = Tr\left[(D^{-1}X\Lambda)^T W (D^{-1}X\Lambda)\right]$

$$- \vec{1}^T W^T (D^{-1}X\Lambda)(D^{-1}X\Lambda)^T W \vec{1}$$

Clustering based on dynamic networks

FacetNet(extensions)

If the number of cluster is change, estimating members of clusters.

$$cost = \alpha \cdot D(W \| X \Lambda X^T) + (1-\alpha) \cdot \underline{D(Z \| X \Lambda X^T)}$$

$$Z \doteq X_{t-1} \Lambda_{t-1} X_{t-1}^T$$

Compare ***structure of networks*** instead of community structure.

$$x_{ik} \leftarrow x_{ik} \cdot \sum_{j} \frac{(\alpha \cdot w_{ij} + (1-\alpha) \cdot z_{ij}) \cdot \lambda_k \cdot x_{jk}}{(X \Lambda X^T)_{ij}}$$

$$\textit{then normalize such that } \sum_{i} x_{ik} = 1, \forall k$$

$$\lambda_k \leftarrow \lambda_k \cdot \sum_{ij} \frac{(\alpha \cdot w_{ij} + (1-\alpha) \cdot z_{ij}) \cdot x_{ik} \cdot x_{jk}}{(X \Lambda X^T)_{ij}}$$

$$\textit{then normalize such that } \sum_{k} \lambda_k = 1.$$

Clustering based on dynamic networks

A Particle-and-Density Based Evolutionary Clustering Method for Dynamic Networks [VLDB’09](Jiawei Han group)——P&D(shorted)

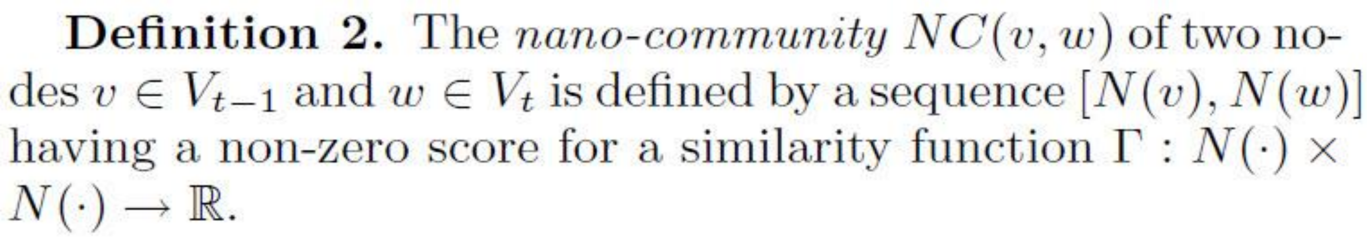
View: observing **objective edges over time** instead of **snapshot** at special time.

Basic Idea: To collect lots of **particles** *(a)* called nano-communities and a community as a **densely connected subset of particles** , called ***l-KK*** *(b)* that guide the trend of evolution. In order to get **local clusters**, a density-based method is proposed by using **optimal modularity** *(c)* .

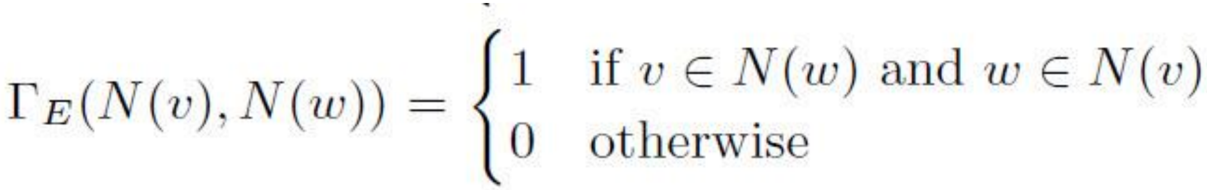
Clustering based on dynamic networks

P&D

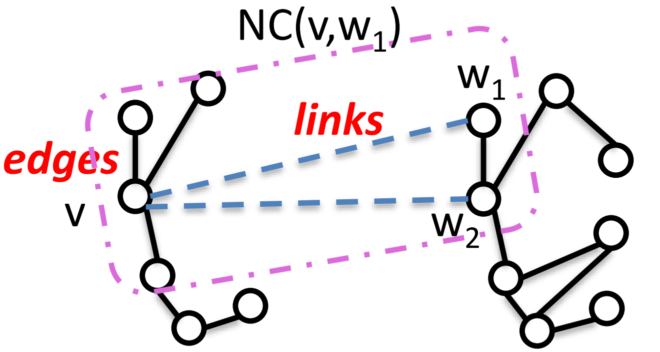
(a) ***nano-communities*** (observing the trend of evolution)



Definition 2. The *nano-community* $NC(v,w)$ of two nodes $v \in V_{t-1}$ and $w \in V_t$ is defined by a sequence $[N(v), N(w)]$ having a non-zero score for a similarity function $\Gamma : N(\cdot) \times N(\cdot) \rightarrow \mathbb{R}$.



$$\Gamma_E(N(v), N(w)) = \begin{cases} 1 & \text{if } v \in N(w) \text{ and } w \in N(v) \\ 0 & \text{otherwise} \end{cases}$$

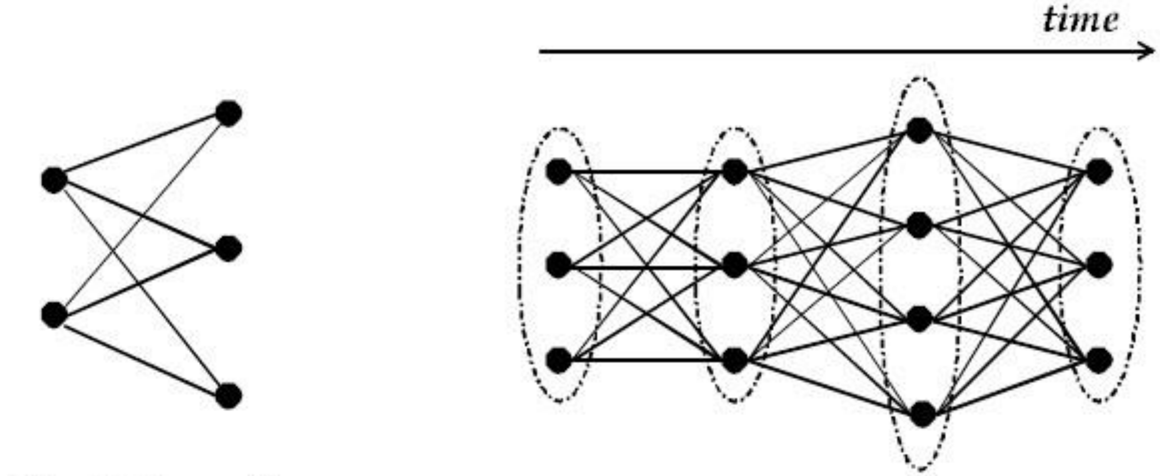


G_{t-1} G_t(current)

Clustering based on dynamic networks

P&D

(b) l-KK



(a) a biclique $K_{2,3}$

(b) a 4-clique by clique $KK_{[3,3,4,3]}$

Definition :

A community of *l-KK* is definitely the densest one among all communities of the same partite sizes.

To measure the similarity between l-KK.

$$\Theta_K(M_t) = \frac{2|E_{M_t}|}{|V_{M_t}|(|V_{M_t}|-1)} \qquad \Theta_{KK}(B_{t-1,t}) = \frac{|L_{B_{t-1,t}}|}{|V_{M_{t-1}}||V_{M_t}|}$$

Clustering based on dynamic networks

P&D

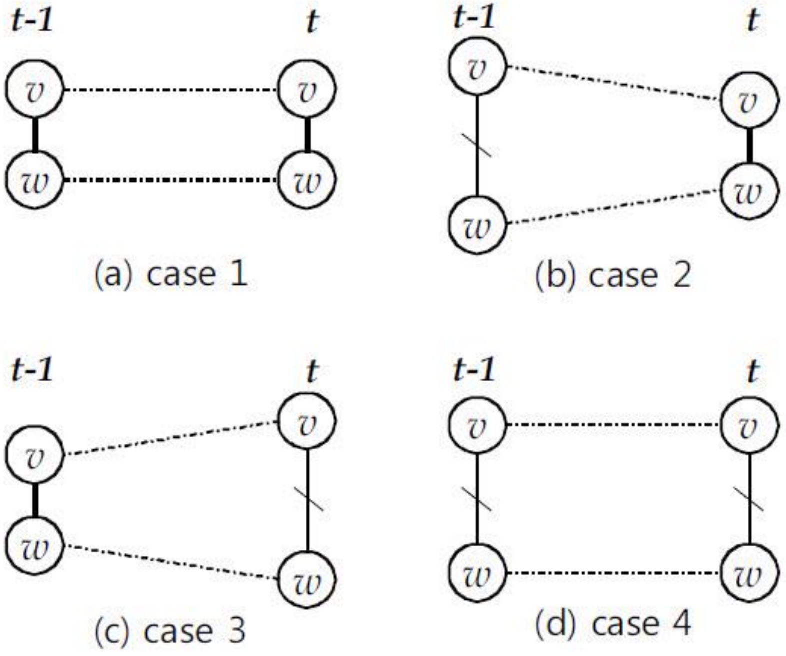
(c) ***clustering(at a snapshot)***

Basic idea: for a snapshot, community structure reflects not only on **current edges**, but also **historic structure**. Because evolution network is **smooth**.

Clustering based on dynamic networks

P&D

(c) clustering(at a snapshot)



(a) case 1 (b) case 2

(c) case 3 (d) case 4

$$d'_t(v,w) = \alpha \cdot \{d_O(v,w) - d_{t-1}(v,w)\} + d_{t-1}(v,w)$$

$$\boxed{d'_t(v,w) = \ \alpha \cdot d_O(v,w) + (1-\alpha)d_{t-1}(v,w)}$$

$d_O(v,w)$ denote distance between nodes at t

Clustering based on dynamic networks

P&D

(c) clustering——density-based clustering

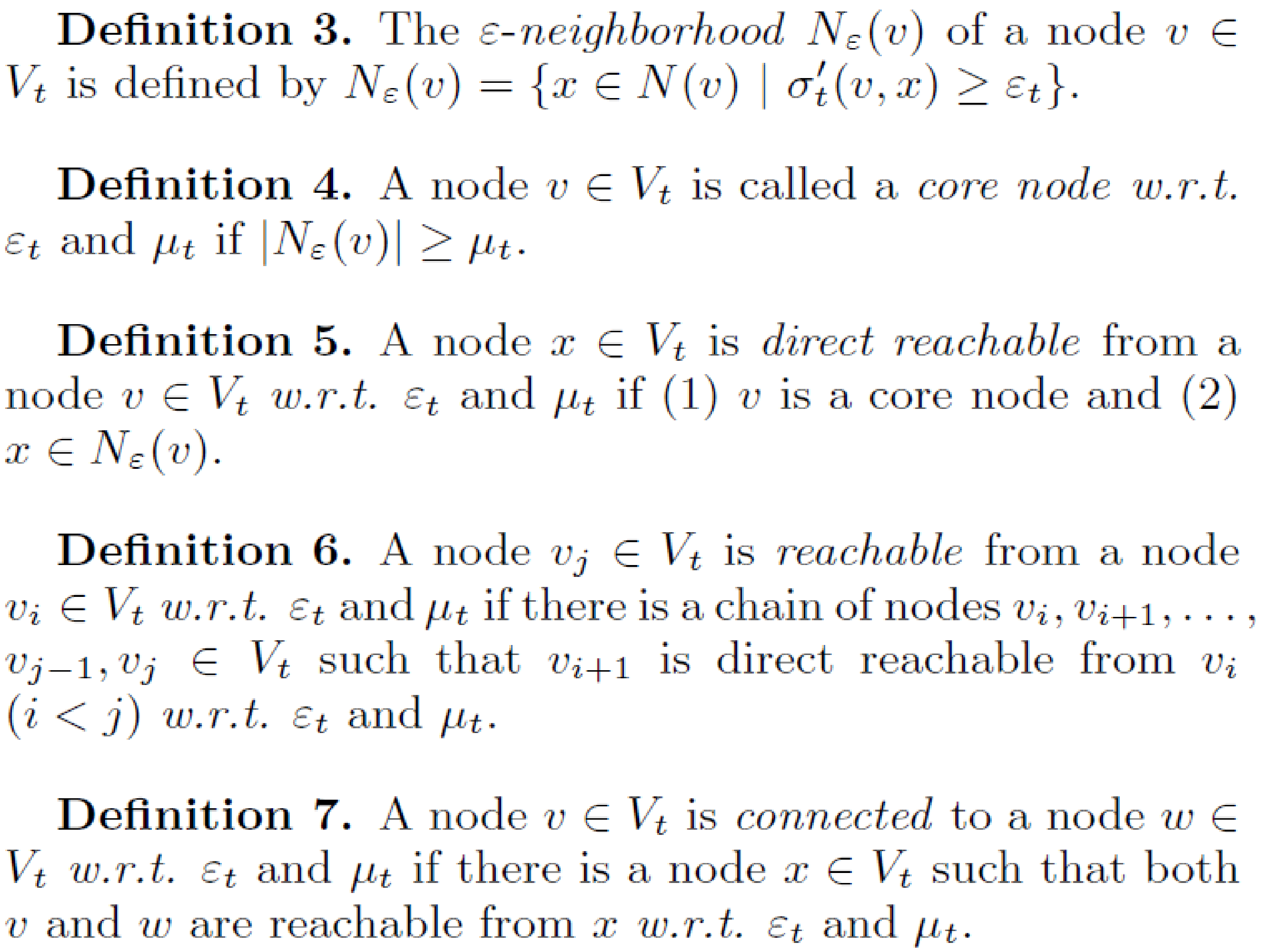
$$d'_t(v,w) = \alpha \cdot \{d_O(v,w) - d_{t-1}(v,w)\} + d_{t-1}(v,w)$$

$$\sigma'_t(v,w) = \alpha \cdot \{\sigma_t(v,w) - \sigma_{t-1}(v,w)\} + \sigma_{t-1}(v,w)$$

$$\sigma(v,w) = \frac{|N(v) \cap N(w)|}{\sqrt{|N(v)| \times |N(w)|}}$$

Clustering based on dynamic networks

DBSCAN



Definition 3. The *ε-neighborhood* $N_\varepsilon(v)$ of a node $v \in V_t$ is defined by $N_\varepsilon(v) = \{x \in N(v) \mid \sigma'_t(v,x) \geq \varepsilon_t\}$.

Definition 4. A node $v \in V_t$ is called a *core node w.r.t.* ε_t and μ_t if $|N_\varepsilon(v)| \geq \mu_t$.

Definition 5. A node $x \in V_t$ is *direct reachable* from a node $v \in V_t$ *w.r.t.* ε_t and μ_t if (1) v is a core node and (2) $x \in N_\varepsilon(v)$.

Definition 6. A node $v_j \in V_t$ is *reachable* from a node $v_i \in V_t$ *w.r.t.* ε_t and μ_t if there is a chain of nodes $v_i, v_{i+1}, \ldots,$ $v_{j-1}, v_j \in V_t$ such that v_{i+1} is direct reachable from v_i $(i < j)$ *w.r.t.* ε_t and μ_t.

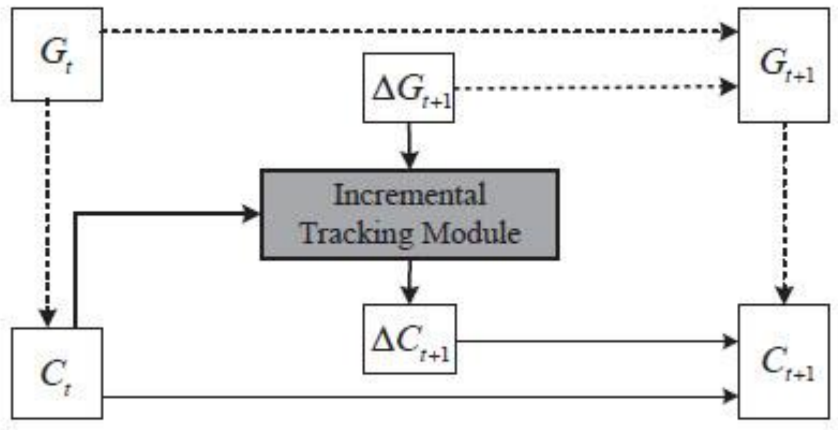
Definition 7. A node $v \in V_t$ is *connected* to a node $w \in V_t$ *w.r.t.* ε_t and μ_t if there is a node $x \in V_t$ such that both v and w are reachable from x *w.r.t.* ε_t and μ_t.

Modularity → Clustering

Clustering based on dynamic networks

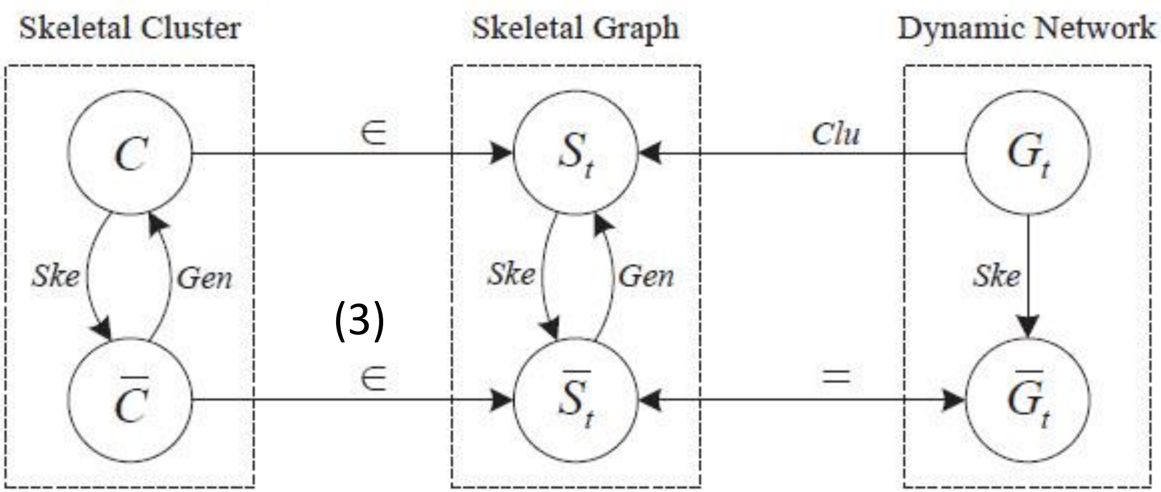
Incremental cluster evolution tracking from highly dynamic network data [ICDE’14]

Basic Idea: Authors want to observe network evolution from subgraph view, which summarized via **skeletal graph** in **fading time window** as time pass.



Clustering based on dynamic networks

Processing:



(1)

(2)

(3)

Clustering based on dynamic networks

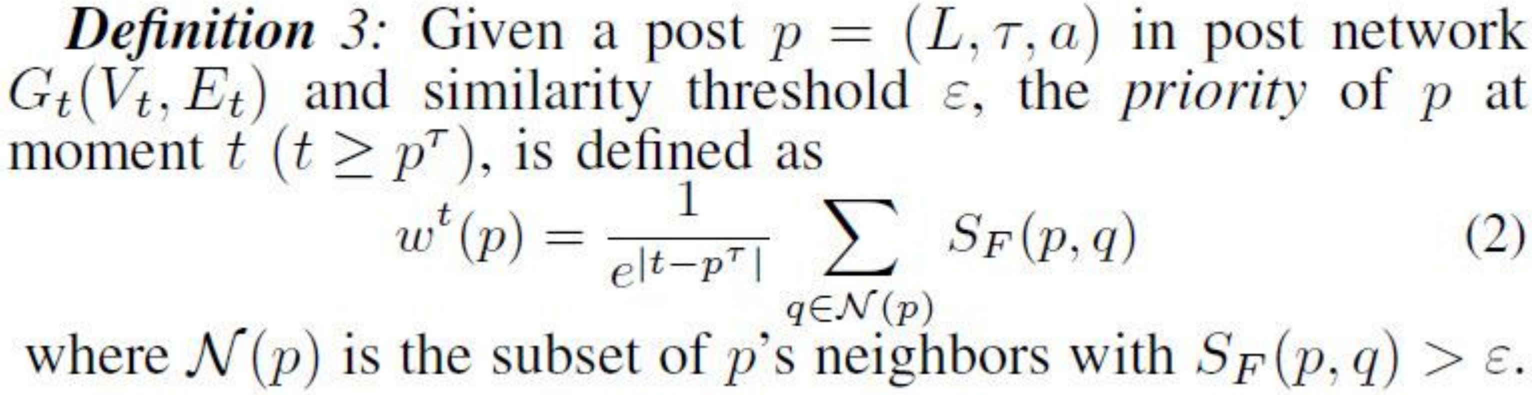
(1) Networks construction(weight of edge)——G_t

$$S_F(p_i, p_j) = \frac{|p_i^L \cap p_j^L|}{|p_i^L \cup p_j^L| \cdot e^{|p_i^\tau - p_j^\tau|}}$$

Where p_i is node i, p_i^L is neighbors of node i, p_i^τ is timestamp.

If $S_F(p_i, p_j) > \lambda$, then build edge.

(2) Skeletal network construction——$\overline{G_t} = Ske(G_t)$



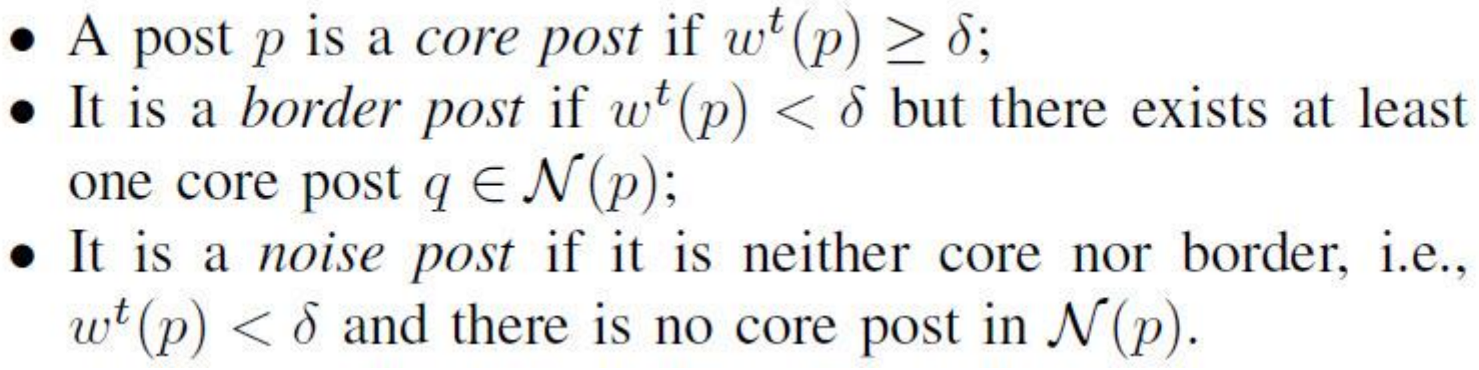
Definition 3: Given a post $p = (L, \tau, a)$ in post network $G_t(V_t, E_t)$ and similarity threshold ε, the *priority* of p at moment t $(t \geq p^\tau)$, is defined as

$$w^t(p) = \frac{1}{e^{|t-p^\tau|}} \sum_{q \in \mathcal{N}(p)} S_F(p, q) \qquad (2)$$

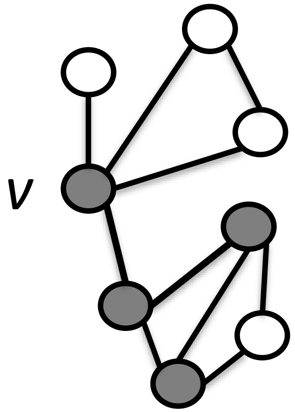
where $\mathcal{N}(p)$ is the subset of p's neighbors with $S_F(p, q) > \varepsilon$.

Clustering based on dynamic networks

Node:

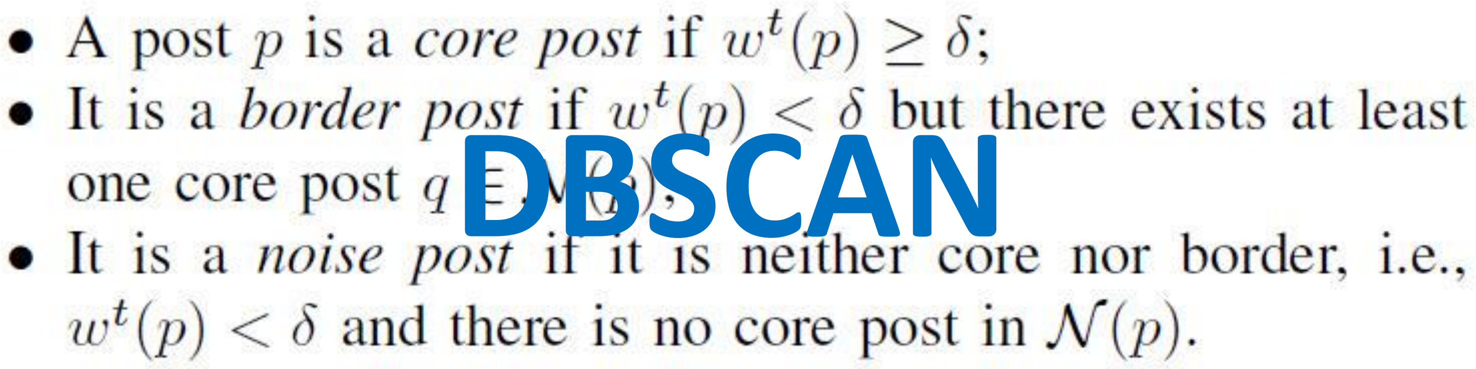


- A post p is a *core post* if $w^t(p) \geq \delta$;
- It is a *border post* if $w^t(p) < \delta$ but there exists at least one core post $q \in \mathcal{N}(p)$;
- It is a *noise post* if it is neither core nor border, i.e., $w^t(p) < \delta$ and there is no core post in $\mathcal{N}(p)$.



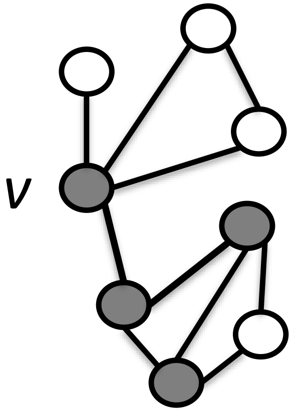
Clustering based on dynamic networks

Node:



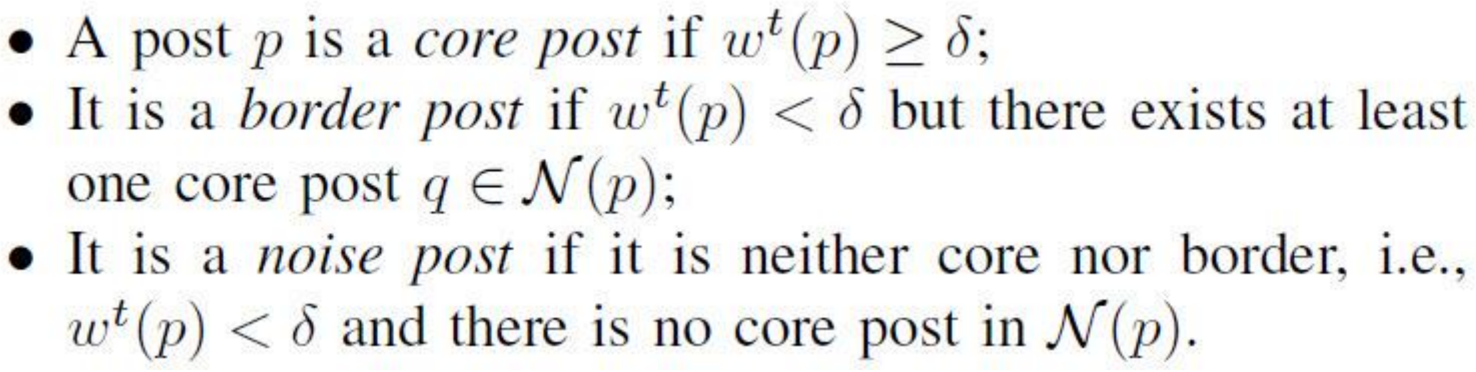
- A post p is a *core post* if $w^t(p) \geq \delta$;
- It is a *border post* if $w^t(p) < \delta$ but there exists at least one core post $q \in \mathcal{N}(p)$;
- It is a *noise post* if it is neither core nor border, i.e., $w^t(p) < \delta$ and there is no core post in $\mathcal{N}(p)$.

DBSCAN



Clustering based on dynamic networks

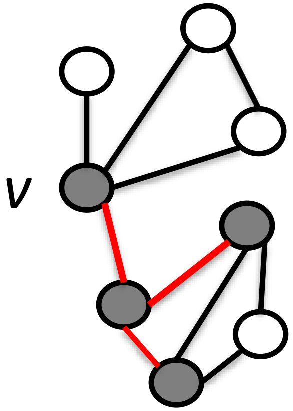
Node:



- A post p is a *core post* if $w^t(p) \geq \delta$;
- It is a *border post* if $w^t(p) < \delta$ but there exists at least one core post $q \in \mathcal{N}(p)$;
- It is a *noise post* if it is neither core nor border, i.e., $w^t(p) < \delta$ and there is no core post in $\mathcal{N}(p)$.

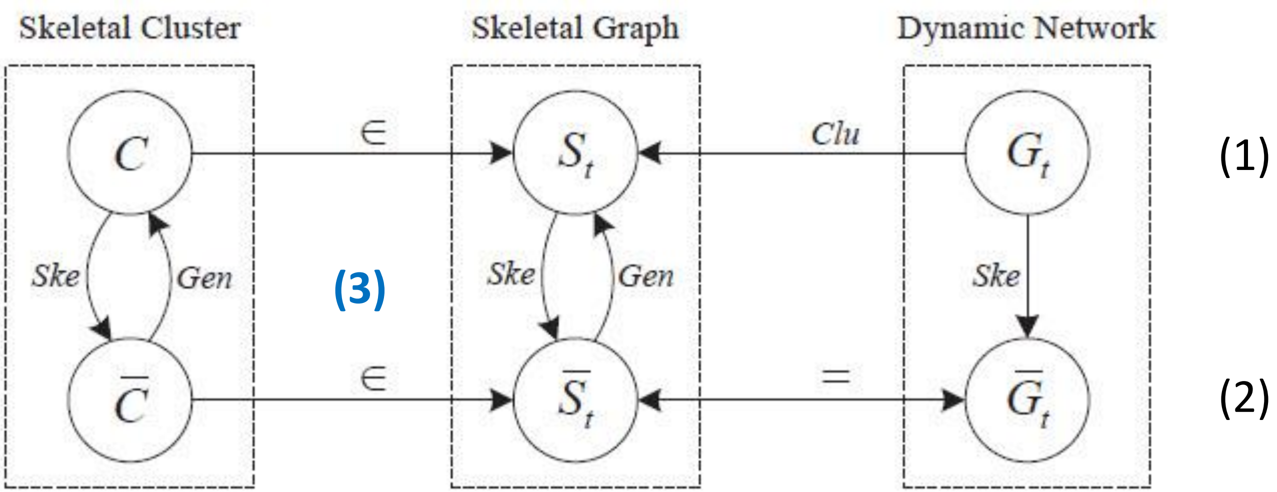
Edge:

Core posts connected by edges with similarity higher than ε will form a summary of $G_t(V_t,E_t)$, that we call the skeletal graph.

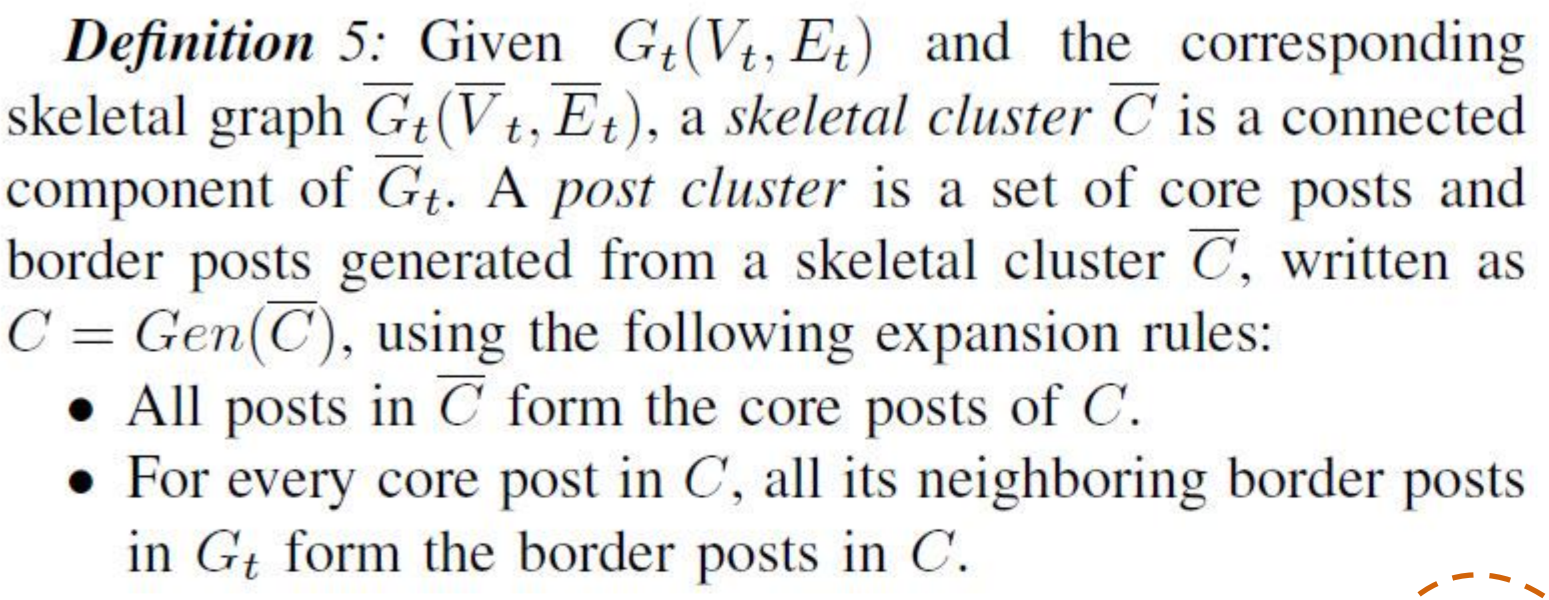


Clustering based on dynamic networks

Processing:

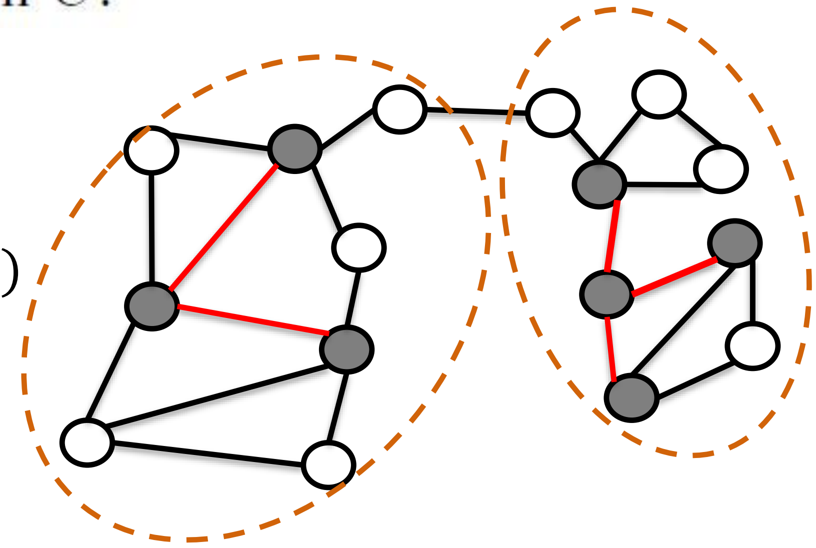


Clustering based on dynamic networks



***Definition** 5:* Given $G_t(V_t, E_t)$ and the corresponding skeletal graph $\overline{G}_t(\overline{V}_t, \overline{E}_t)$, a *skeletal cluster* $\overline{C}$ is a connected component of $\overline{G}_t$. A *post cluster* is a set of core posts and border posts generated from a skeletal cluster $\overline{C}$, written as $C = Gen(\overline{C})$, using the following expansion rules:

- All posts in $\overline{C}$ form the core posts of C.
- For every core post in C, all its neighboring border posts in G_t form the border posts in C.



$\boldsymbol{C = Gen(\overline{C})}$

Clustering based on dynamic networks

Update Networks

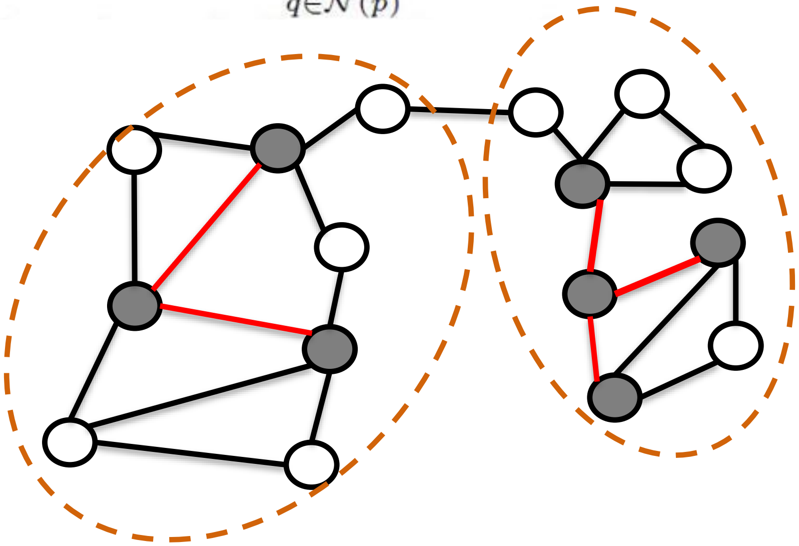
- ✓ Initial network G_0. get $\overline{G_0}$ (skeletal), $\overline{C}$ (cluster skeletal) , C(clusters generated by cluster skeletal).[choose a time window]
- ✓ Add node i, **update G_0,**

$$S_F(p_i, p_j) = \frac{|p_i^L \cap p_j^L|}{|p_i^L \cup p_j^L| \cdot e^{|p_i^\tau - p_j^\tau|}}$$

$\overline{G_0}$

$$w^t(p) = \frac{1}{e^{|t-p^\tau|}} \sum_{q \in \mathcal{N}(p)} S_F(p, q)$$

$\overline{C}$, C :



- ✓ ***Capture network evolution***

Clustering based on dynamic networks

Dynamic Community Detection in Weighted Graph Streams
[ICDM’13](Philip S. Yu group)

Local Weighted-Edge-based Pattern(LWEP)

Basic Idea: In order to detect communities under network stream, there arrange two parts to handle the task: online and offline.

Online component : maintains the statistics top-k neighbors and top-k candidate that identify activity of nodes.
[**capture real-time information**]

Offline component : find local pattern based on top-k neighbors lists.
[**identify patterns**]

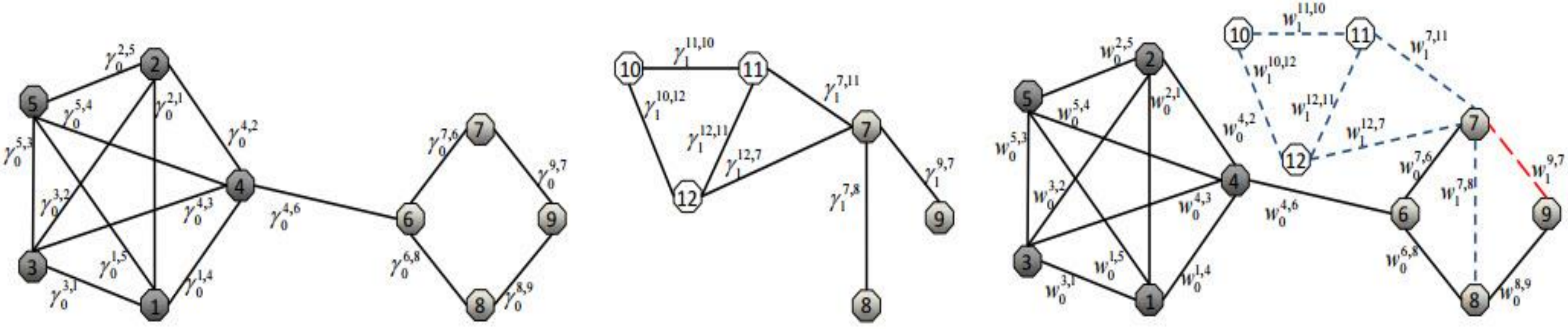
Method of clustering : maintain **small clusters(Local Weight-Edge-based Pattern, LWEP)** based on similarity and further, **clustering** small clusters.

Clustering based on dynamic networks

- In order to detect clusters, authors want to **find local pattern** (density) and **then clustering** those local dense area.
- In order to maintain those local dense area, authors hope to **maintain edges with the highest weights** based on dynamic graph.
- How to maintain those edges with the highest weights?

Clustering based on dynamic networks

Scenario



(a) Initial graph $\overline{\mathcal{G}}_0$ (b) Incremental graph $\overline{\mathcal{G}}_1$ (c) Accumulated graph $\mathcal{G}_1$

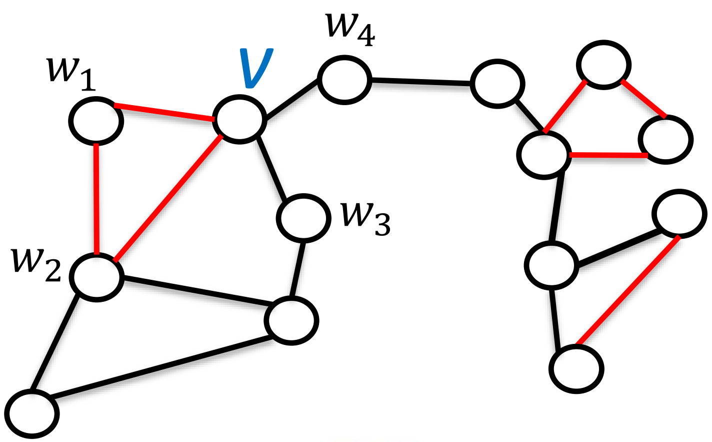
Basic structure :

Weight of edges(decay function)

$$w_t^{i,j} \leftarrow w_{t_e}^{i,j} e^{-\lambda(t-t_e)} + \gamma_t^{i,j}$$

Clustering based on dynamic networks

Similarity between nodes:



Jaccord distance with weighted

Threshold

$$WT(v^i) \triangleq \frac{\sum_{v^j \in \mathcal{WN}(v^i)} WJN(\{v^i, v^j\})}{|\mathcal{WN}(v^i)|}$$

Get local pattern, called (LWEPs)

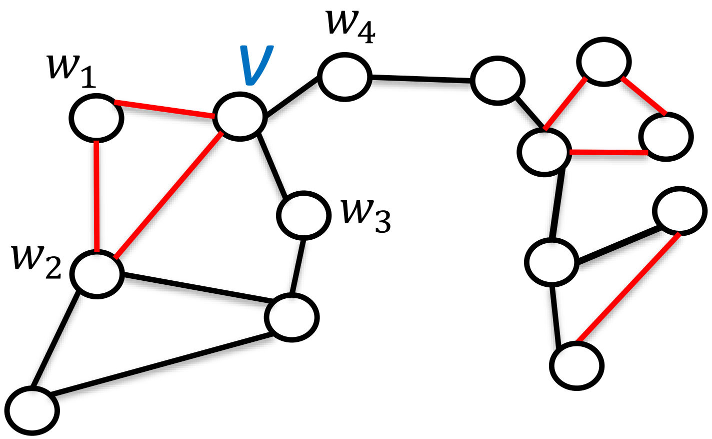
华丽的分割线

How to maintain LWEPs in the graph stream processes?

Clustering based on dynamic networks

Top-k nodes selection

Main Idea: the computation of LWEPs mainly relies on **the neighbors with the highest weights**. If we can **preserve** a small **set of highly weighted neighbors**, no significant information will be lost. That is, we have the following neighbor selection principle.



For node v :

Top-k neighbors:

$w_2, w_1;$

Clustering based on dynamic networks

Question

If just store top-k neighbors with the highest weights, new edges will be **lost** when its weight is less than all top-k neighbors’.

Top-*k* neighbors

Store **the neighbors with the highest weights.**

Top-*k* candidates

Store **candidates neighbors with weight less top-k’s.**

Over time

In order to decrease memory.

Introduce time window.

Clustering based on dynamic networks

Local structure detection (LWEP)

Threshold $WT(v^i) \triangleq \frac{\sum_{v^j \in \mathcal{WN}(v^i)} WJN(\{v^i, v^j\})}{|\mathcal{WN}(v^i)|}$

For each node: $w^{i,i} = \max_{(v^j, w^{i,j}) \in \mathcal{WN}(v^i)} \{w^{i,j}\}$

Which edges need to count?

Pick out edges with high weight form **top-k neighbors and top-k candidates** after updating.

Cluster generation —— merge local patterns(similarity)

Over!!!

Summery

FacetNet

P&D

XXX(increasing)

XXX(graph stream)

More feasible

More fast

More …

Thanks

Zhongjing Yu
yuzhongjing@foxmail.com