A Bottom-up Hierarchical Approach for Joint Inference of Overlapping Social Circles and Missing User Information

Carnegie Mellon University

Vignesh Kannan, Siddharth Kannan

MACHINE LEARNING DEPARTMENT

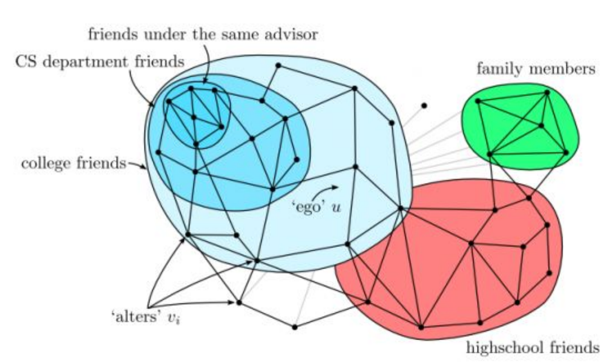
Motivation

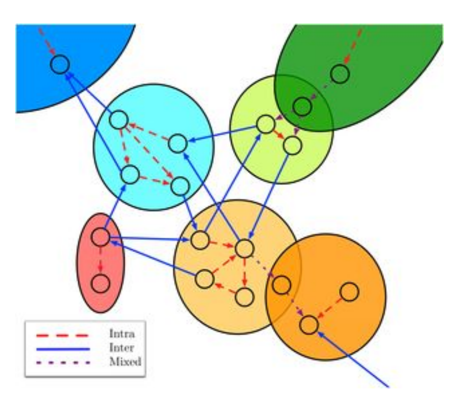
- Natural to think of social networks as being composed of many interconnected social circles
- Filtered sharing of content as decided by the user
- Strategic marketing and advertisements targeted at particular groups leading to higher financial returns
- Identify anomalies and terrorist connections in social circles for purposes of international security
- Infer missing information about users from their memberships to different social circles and improve friend and page suggestions
- Facilitate the automated formation of online communities and forums to mobilize support for various causes.

Properties of Social Circles

- Social circles evolve in a hierarchical bottom-up fashion.
- Circles can overlap, or in other words, a user can belong to more than one circle.
- Strong circles can form within weak circles.
- Principle of homophily is captured by the framework.
- Users with many mutual friends are likely to be connected.

Principle of homophily: *People who have similar attributes tend to be connected, and people who are conneced tend to have similar attributes.*





Contribution

- A **nonparametric (model-free) algorithm** that is not limited by assumptions about data distribution that entail parametric approaches
- Ability to naturally **infer the number of social circles** without requiring it as input and therefore, very useful from a practical standpoint
- Uses both **graph structure information and node attribute information**
- Inbuilt mechanism to **produce hierarchical, disjoint and overlapping circles**
- Effective strategy to **eliminate redundant circles** formed
- **Inference of potentially missing information** in the user profile and thereby useful for recommending links between users

Dataset Description

Stanford Network Analysis Project (SNAP), Social circles: Facebook

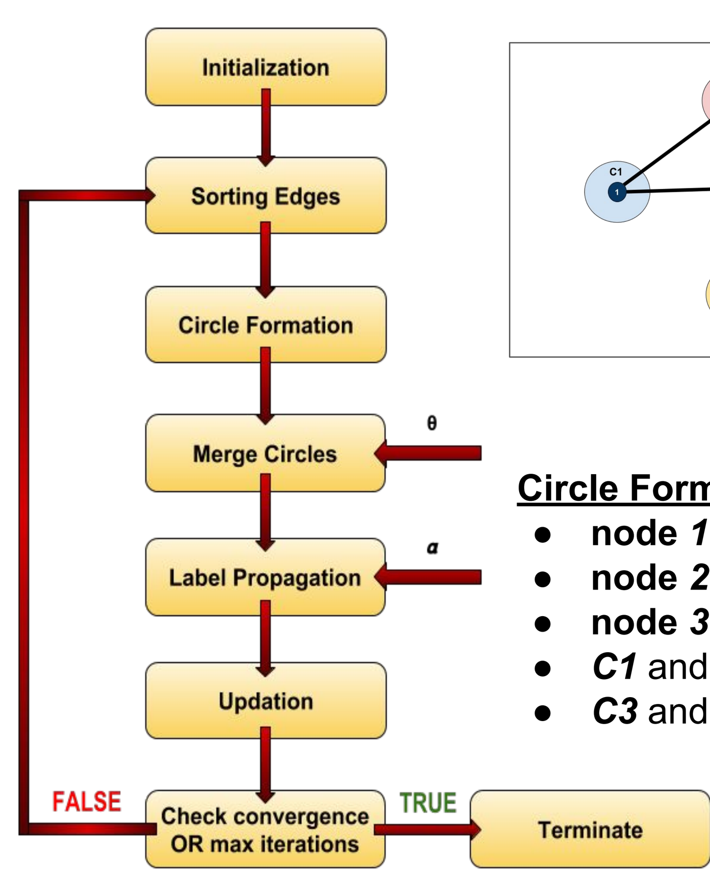
- Number of nodes: 4039
- Number of edges: 88234
- Average clustering coefficient: 0.6055
- Number of ego-networks: 10
- Anonymized feature values obscuring interpretation of features
- Each feature is binary-valued

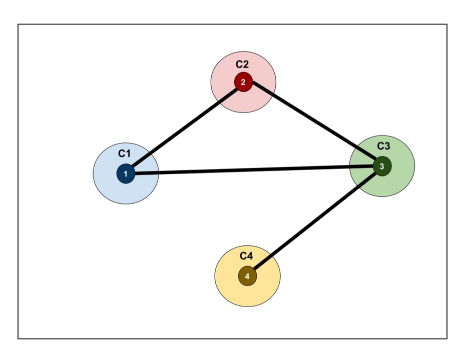
Approach

Algorithm 1

procedure MAIN(θ, α, T)
  Number the friends of the given ego from 1 to n
  Define $\mathcal{C} = \{c_i | 1 \leq i \leq n\}$
  Define $\mathcal{J} = \frac{1}{n}\sum_{i=1}^{n} |x_t^{(i)} - x_{t+1}^{(i)}|$, where $x_t^{(i)}$ is the attribute vector of node i at the t^{th} iteration
  Define $\mathcal{V} = \{1, 2, ...n\}, \mathcal{E} = \{(i,j)|$ i and j are friends$\}$
  for $c_i \in \mathcal{C}$ **do**
    $c_i.members = i$
  $t \leftarrow 0$
  while $t \leq T$ **do**
    CIRCLEFORMATION($\mathcal{V}, \mathcal{E}, \mathcal{C}$)
    MERGECIRCLES($\mathcal{V}, \mathcal{C}, \theta$)
    LABELPROPAGATION($\mathcal{V}, \mathcal{C}, \alpha$)
    Terminate if $\mathcal{J} < \epsilon$
    $t \leftarrow t + 1$
procedure CIRCLEFORMATION($\mathcal{V}, \mathcal{E}, \mathcal{C}$)
  for $(u, v) \in \mathcal{E}$ **do** ▷ edges in $\mathcal{E}$ are sorted in descending order of similarity
    for $i \in$ u.membership / v.membership **do**
      ADDTOCIRCLE(c_i, v) ▷ ADDTOCIRCLE(c_i, v) adds v to $c_i.members$ and c_i to v.membership depending on the similarity of v to all the members of c_i
    for $i \in$ v.membership / u.membership **do**
      ADDTOCIRCLE(c_i, u)
procedure MERGECIRCLES($\mathcal{V}, \mathcal{C}, \theta$)
  for $c_i \in \mathcal{C}$ **do**
    for $c_j \in \mathcal{C}$ **do**
      if IoU(c_i, c_j) $> \theta$ **then**
        for k $\in c_j.members$ **do**
          Remove c_j from k.membership
          Add c_i to k.membership
          Add k to $c_i.members$
        Delete c_j from $\mathcal{C}$
procedure LABELPROPAGATION($\mathcal{V}, \mathcal{C}, \alpha$)
  for $i \in \mathcal{V}$ **do**
    $neighbors = \{j \in \mathcal{V} | j \in c.members, \forall c \in i.membership\}$
    $x_{new}^{(i)} = \frac{\sum_{k \in neighbors} similarity(k,i) x^{(k)}}{\sum_{k \in neighbors} similarity(k,i)}$
    $x^{(i)} \leftarrow \alpha x^{(i)} + (1 - \alpha) x_{new}^{(i)}$

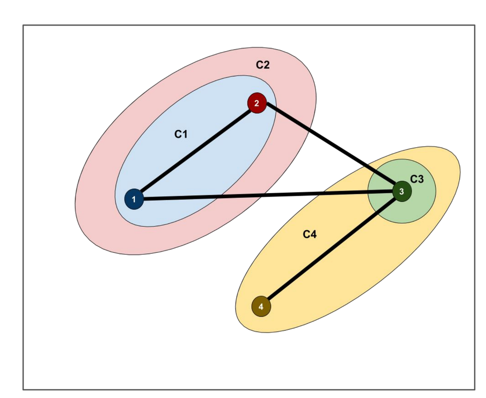
Time complexity: O(T(ElogE + V^2(|max_members| + |max_membership|) + E|max_membership||max_members|))

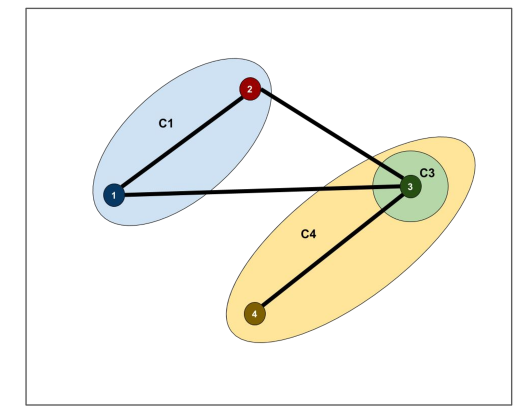




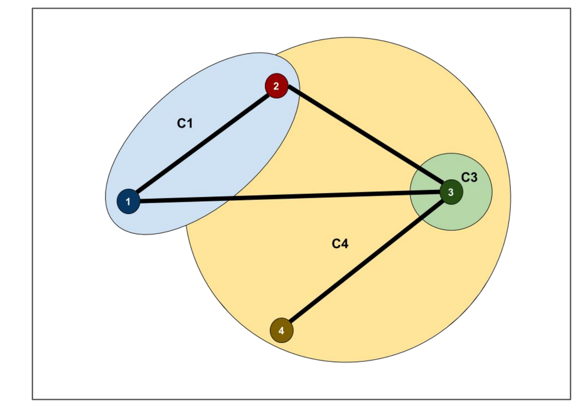
Initialization and Sorting:
- Initialize each node as a cluster. Hence **node *i*** is ***Ci***.
- Sort the edges in decreasing order of **cosine-similarity** between attribute vectors of edge's nodes.

Circle Formation - Iteration 1:
- **node *1*** is assigned to ***C2***.
- **node *2*** is assigned to ***C1***.
- **node *3*** is assigned to ***C4***.
- ***C1*** and ***C2*** overlap.
- ***C3*** and ***C4*** overlap.





Merge Circles and Label Propagation:
- ***C1*** and ***C2*** have high **IoU** (overlap).
- ***C1*** and ***C2*** are merged into a single circle.
- Attributes propagate among circles.



Circle Formation - Iteration 2:
- **node *2*** is assigned to ***C4***.
- ***C1*** and ***C4*** overlap.
- ***C3*** and ***C4*** overlap.

Results

Method		Average 1 - BER	Average F1-score
Link Type	**Similarity**		
Average Link	Cosine	0.6349	0.3112
Complete Link	Cosine	0.7019	0.3832
Average Link	Gaussian Kernel	0.6223	0.4247
Complete Link	Gaussian Kernel	**0.7158**	**0.4667**
McAuley et al. 2012		0.8400	0.5900

Sample performance of Complete Link + Gaussian Kernel on individual egonets

Egonet ID	1 - BER	F1-score
348	0.6687	0.3565
414	0.7264	0.4838
686	0.5305	0.5328
698	0.8877	0.7379

Sample performance of Complete Link + Cosine on individual egonets

Egonet ID	1 - BER	F1-score
348	0.6601	0.3686
414	0.7622	0.5165
686	0.6187	0.3461
698	0.7228	0.3982

Conclusion and Future Work

- Complete link better captures the formation of social circles than Average link, which is more aggressive than desired
- Gaussian kernel yields a better F1-score than Cosine similarity on an average
- Allowing label propagation promotes the formation of bigger circles in the ground truth

- Establishing interpretability of circles formed
- Need to account for one-hot features separately by normalizing them in Label Propagation step
- Work on theoretical convergence
- Experiment with other datasets lilke Google+ and Twitter