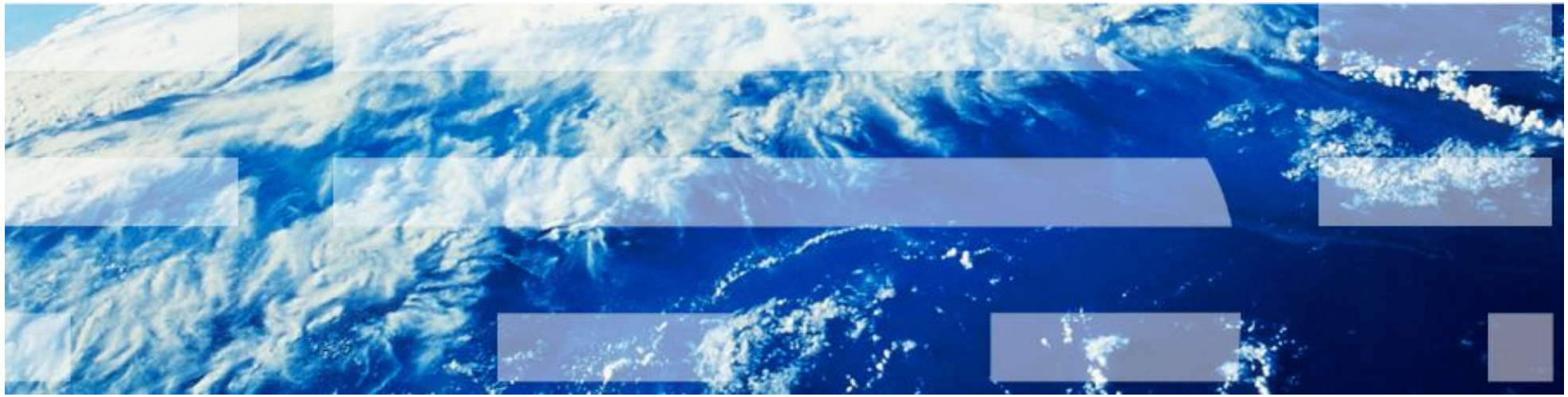
E6893 Big Data Analytics:

***201812-1 Group***
***Community Detection in Anti-Money Laundering (AML)***

**Team Members (with UNI):**
**Siyu Liu(sl4262), Tianyao Hua(th2706), Anke Xu(ax2127)**

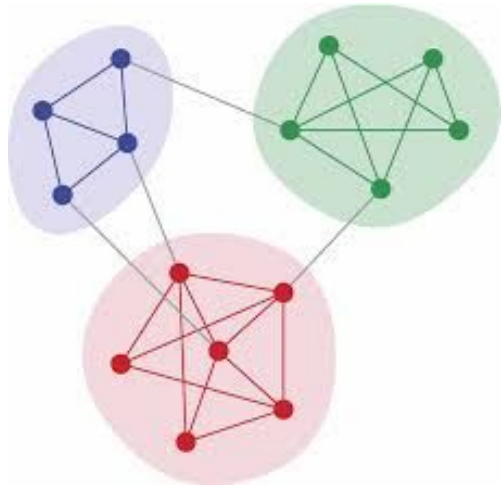


# Content

- Motivation
- Algorithm
  - Pros
  - Cons
  - Modified
- Visualization & Results

# Motivation

- **The Challenging Money Laundering Issue**
  - Groups of collaborating individuals
  - Numerous transactions
  - Offshore accounts and complex investment vehicles with well-connected transaction behaviours



- **The Effectiveness of Community Detection in AML**
  - Overcome the problems of focusing on individuals
  - Consider collective behaviour for each entities and transfer amount information at the same time

- **The Objectives of this Project**
  - Detect suspicious and well-connected entities using CESNA algorithm
  - Visualize the graphical data of financial transfers and results of community detection

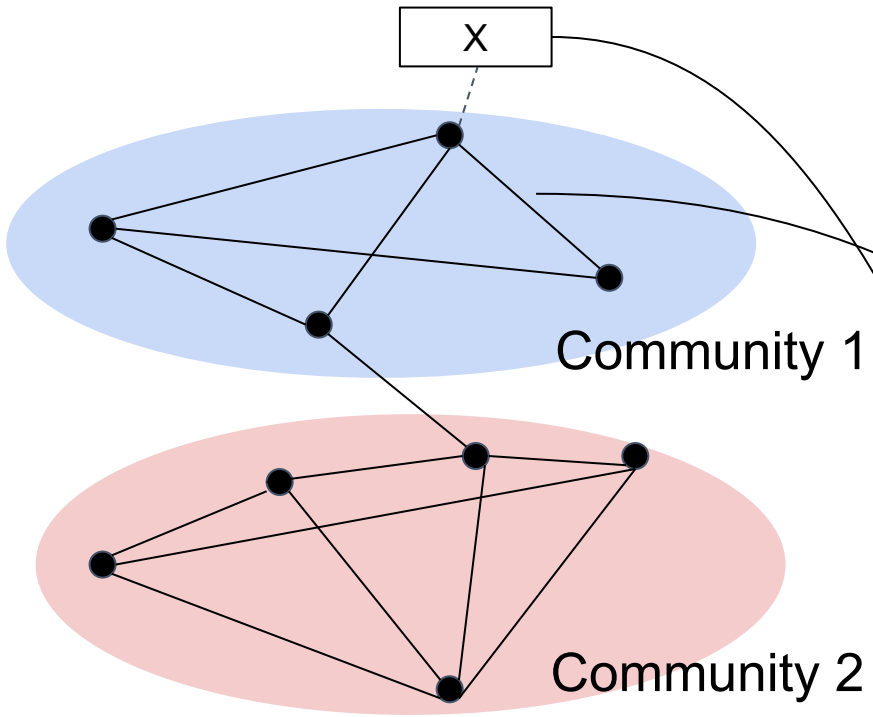
# Algorithm: CESNA
## ----Pros

| Method class | Overlapping | Hard membership | Structure+Attributes | nodes per 10 hours |
|---|---|---|---|---|
| Heuristic | No | Yes | No | 100,000 |
| LDA-based | Yes | No | Yes | 85,000 |
| Clique-based | Yes | Yes | No | 100,000 |
| Social circles | Yes | Yes | No | 5,000 |
| **CESNA** | Yes | Yes | Yes | **1,000,000** |

246 Citations!

Ref: Yang, Jaewon & McAuley, Julian & Leskovec, Jure. (2014). **Community Detection in Networks with Node Attributes**. *Proceedings - IEEE International Conference on Data Mining, ICDM*. 10.1109/ICDM.2013.167.

# Algorithm: CESNA
## ----Cons




Probabilistic Network Model

Model

- Network denoted by $G(V, E)$, has $C$ comm
- Each
- Each
- Netw

Does not support weights!

$F_{uc}\}$

$$P_{uv} = 1 - e^{(-\Sigma_c F_{uc} F_{vc})}$$

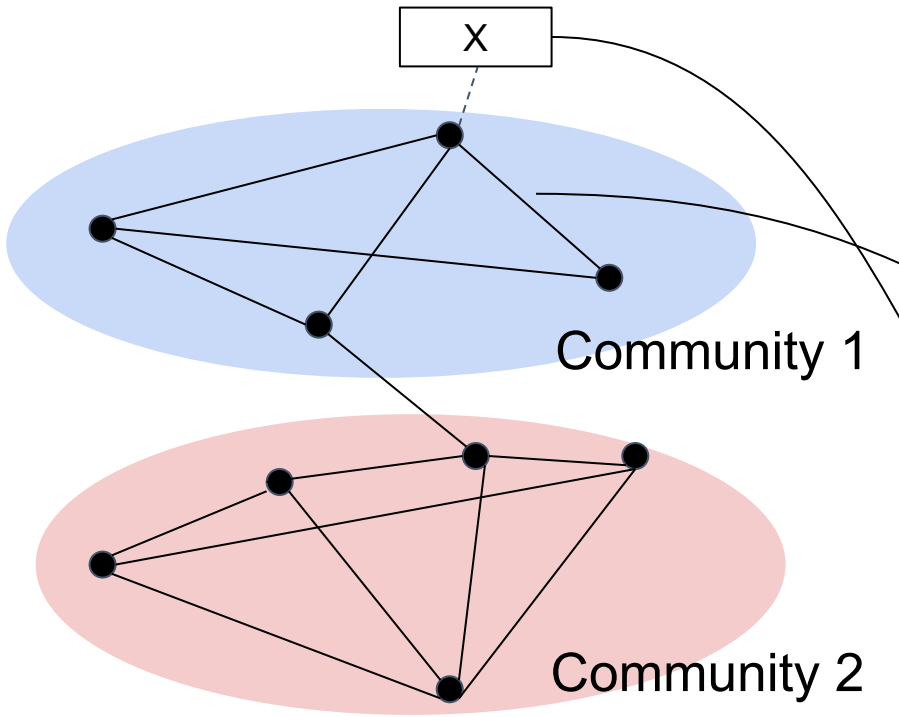
$$A_{uv} \sim \text{Bernoulli}(P_{uv})$$

- Attributes

$$Q_{uk} = \frac{1}{1+e^{-\Sigma_c W_{kc} F_{uc}}}$$

$$X_{uk} \sim \text{Bernoulli}(Q_{uk})$$

# Algorithm: CESNA
## ----Modified




Model

- Network denoted by $G(V, E)$, has $C$ communities
- Each node has attributes $X$, a vector
- Each node has affiliation weights $\{F_{uc}\}$
- Network Weight Matrix

$$\mu_{uv} = \sum_c F_{uc} F_{vc}$$

- Attribu

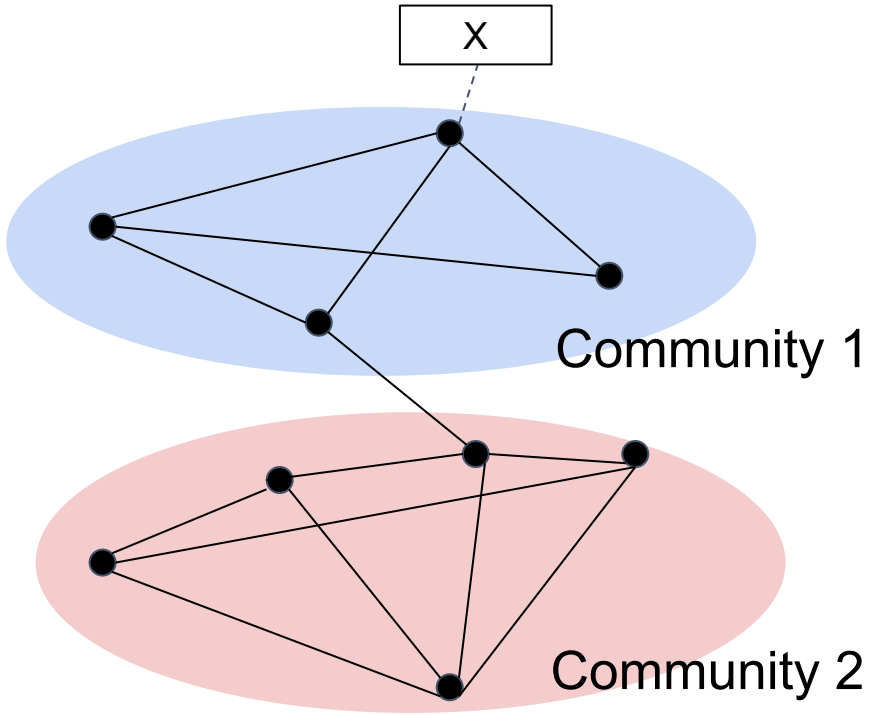
$$W_{uv} \sim Gaussian(\mu_{uv}, \sigma^2)$$

$$Q_{uk} = \frac{1}{1+e^{-\Sigma_c W_{kc} F_{uc}}}$$

$$X_{uk} \sim \text{Bernoulli}(Q_{uk})$$

Probabilistic Network Model

# Algorithm: CESNA
## ----Modified




Probabilistic Network Model

Maximize likelihood to estimate $F$ and $W$

$$\hat{F}, \hat{W} = \underset{F \geq 0, W}{\operatorname{argmax}} \log P(G, X | F, W)$$

Where

$$\log P(G, X | F, W) = L_G + L_X$$

$$L_G = \log P(G | F)$$

Modified

$$L_X = \log P(X | F, W)$$

# Algorithm: CESNA ----Modified

**Updating community memberships.** To update community memberships, we use a similar optimization procedure used in BigCLAM [41]. However, we modify the procedure to consider node attributes (which BigCLAM ignores). We update the membership $F_u$ of an individual node $u$ while fixing all other parameters (the membership $F_v$ of all other nodes, and logistic model parameters $W$).

We solve the following subproblem for each $u$:

$$\hat{F}_u = \underset{F_{uc} \geq 0}{\operatorname{argmax}} \mathcal{L}_G(F_u) + \mathcal{L}_X(F_u), \quad (5)$$

where $\mathcal{L}_G(F_u)$ and $\mathcal{L}_X(F_u)$ are the parts of $\mathcal{L}_G, \mathcal{L}_X$ involving $F_u$, *i.e.*,

$$\mathcal{L}_G(F_u) = \sum_{v \in \mathcal{N}(u)} \log(1 - \exp(-F_u F_v^T)) - \sum_{v \notin \mathcal{N}(u)} F_u F_v^T$$

$$\mathcal{L}_X(F_u) = \sum_k (X_{uk} \log Q_{uk} + (1 - X_{uk}) \log(1 - Q_{uk}))$$

where $\mathcal{N}(u)$ is a set of neighbors of $u$. Note that this problem is convex: $\mathcal{L}_G(F_u)$ is a concave function of $F_u$ [41], [30] and $\mathcal{L}_X(F_u)$ is a logistic function of $F_{uc}$ when $W$ is fixed.

To solve this convex problem, we use projected gradient ascent. The gradient can be computed straightforwardly:

$$\frac{\partial \mathcal{L}_G(F_u)}{\partial F_u} = \sum_{v \in \mathcal{N}(u)} F_{vc} \frac{\exp(-F_u F_v^T)}{1 - \exp(-F_u F_v^T)} - \sum_{v \notin \mathcal{N}(u)} F_{vc}$$

$$\frac{\partial \mathcal{L}_X(F_u)}{\partial F_u} = \sum_k (X_{uk} - Q_{uk}) W_{kc}.$$

$$L_G(F_u) = \sum_{u \neq v} \log \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{||W_u - \mu_u||^2}{2\sigma^2}}$$

?? (on $u \neq v$)

$$\frac{\partial L_G}{\partial F_u} = \sum_v (W_{uv} - \mu_{uv}) F_v - (W_{uu} - \mu_{uu}) F_u$$

⇩ gradient ascent

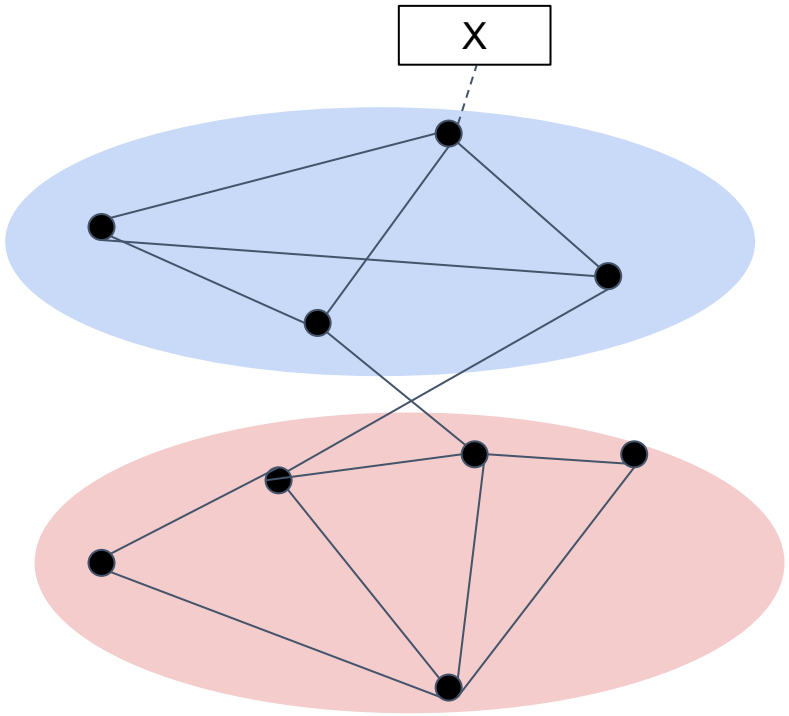
$$F_{uc}^{new} = \max(0, F_{uc}^{old} + \alpha(\frac{\partial \mathcal{L}_G(F_u)}{\partial F_u} + \frac{\partial \mathcal{L}_X(F_u)}{\partial F_u}))$$

$$W_{kc}^{new} = W_{kc}^{old} + \alpha(\sum_u \frac{\partial \log P(X_{uk}|F, W_k)}{\partial W_{kc}} - \lambda \cdot \text{Sign}(W_{kc}))$$

*l*1 regularized, select features.

# Algorithm: CESNA ----Modified

- **Algorithm-CESNA**




Probabilistic Network Model

Determine Membership

- $u$ belongs to $c$ if $F_{uc} > \delta$
- Choose threshold?

$$\frac{1}{N} < 1 - e^{-\delta^2}$$

What's the number of $C$?

- Reserve a holdout set

# Visualization & Results

# Appendix - Simulated Dataset

Based on the fraud detection data on Kaggle competition, we performed bootstrap, modification, and simulation according to the basic logics we defined for money laundering activities. We generated two dataset based on the following rules. (https://www.kaggle.com/netzone/eda-and-fraud-detection/data)

```
1 data.head()
```

| | step | type | amount | nameOrig | oldbalanceOrg | newbalanceOrig | nameDest | oldbalanceDest | newbalanceDest | isFraud | isFlaggedFraud |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | PAYMENT | 9839.64 | C1231006815 | 170136.0 | 160296.36 | M1979787155 | 0.0 | 0.0 | 0 | 0 |
| 1 | 1 | PAYMENT | 1864.28 | C1666544295 | 21249.0 | 19384.72 | M2044282225 | 0.0 | 0.0 | 0 | 0 |

Dataset 1:

1. Bootstrap
2. Simulate communities
   a. Randomly set the size of a community
   b. Randomly select n ids within non-fraud/fraud data to form groups
3. Simulate reasonable node attributes
   a. Occupation, b. Account open country, c. Account type, d. Amount Modification

Dataset 2:

1. Set different number of communities and nodes
2. Simulate connections between and within communities
3. Simulate node attributes