

# **SIGMOD2013 勉強会**

## **Session 8: Social Media**

筑波大 山口

# session 8:

## social media

- ✓ **Efficient sentiment correlation for large-scale demographics**
  - ✓ Tsytsarau et al. (Univ. of Trento)
- ✓ **EBM - An Entropy-Based Model to Infer Social Strength from Spatiotemporal Data**
  - ✓ Pham et al. (Univ. of Southern California)
- ✓ **Online Search of Overlapping Communities**
  - ✓ Cui et al. (Fudan Univ.)

# Efficient Sentiment Correlation for Large-scale Demographics

13/09/07

SIGMOD勉強会 - Yuto Yamaguchi



# Problem

グループごとのsentiment analysisをする研究は少ない

✓ 例) イタリアの学生の意見、ヨーロッパの10代の意見

## 扱う問題

- ✓ ある期間において  
あるトピックに対するsentimentが似ているグループを探す
- ✓ 例) 小学生女子「プリキュア大好き！」  
理系大学生男子「プリキュア大好き！」

# Data

扱うデータは次のタプルの集合

- ✓  $x = (u, t, s, p)$
- ✓  $u$ : ユーザ
- ✓  $t$ : トピック
- ✓  $s$ : センチメント  $s \in [-1, 1]$  (1に近いほどプラスの感情)
- ✓  $p$ : 期間 (2013年9月の一ヶ月間とか)
- ✓  $x =$  (誰がいつ何に対してどんな感情を示した)
  
- ✓ 例)  $x_1 = ('Alice', 'Politics', 0.8, p_1)$   
 $x_2 = ('Bob', 'Drama', -0.5, p_2)$

# Demographics

それぞれのユーザは Demographic Criteria  $d$  をもつ

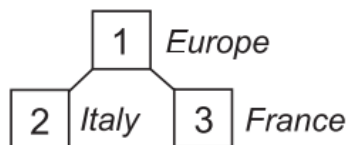
✓ 例)  $d = \{\text{age: Young, location: Italy, occupation: Student}\}$

Demographicは階層構造を持つ

✓ 例)  $\text{Europe} = \{\text{Italy, France}\}$ ,  $\text{Academic} = \{\text{Prof., Student}\}$

Demographic Criteria  $d$  は階層構造に従ってLatticeになる

hierarchy 1: location



hierarchy 2: occupation



demographics lattice

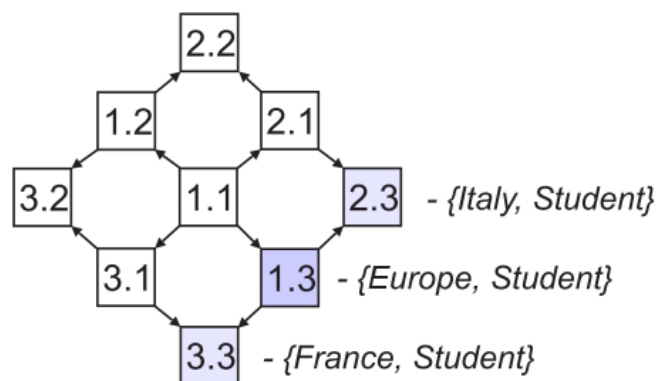


Figure 1: Two demographics hierarchies forming a lattice.

# Sentiment Series

## Demographic Group $U_d$

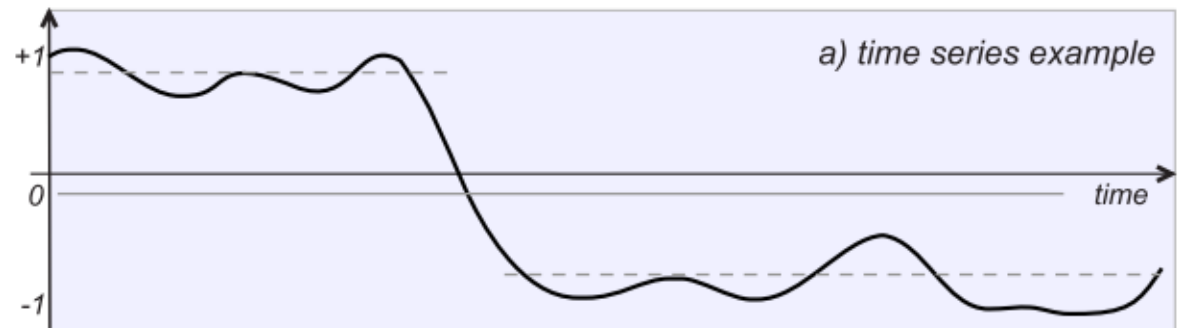
- ✓ 同じ Demographic Criteria  $d$  を持つユーザの集合

## Group Sentiment $s_i$

- ✓ あるトピック  $t$ 、ある時刻  $p$  におけるグループのsentimentの平均

## Sentiment Series

- ✓  $s_i$  の列



# Correlated Sentiment

Sentiment Correlation (ピアソンの積率相関係数)

✓ 二つのSentiment seriesがどれだけ似ているか

Correlated Sentiment(本論文の問題)

✓ あるトピック  $t$ 、ある期間  $p$  に対して  
相関係数が閾値以上となるDemographic Groupのペアを返す



# **EBM - An Entropy-Based Model to Infer Social Strength from Spatiotemporal Data**

13/09/07

SIGMOD勉強会 - Yuto Yamaguchi



# Problem

LBSN(Location-based social network)

✓ Foursquare, Facebook, Twitter, ...

## 問題

✓ チェックイン履歴を用いて友人関係の強さを推定

## 仮説

✓ 同じ場所に同時にチェックイン(共起)する二人は友達っぽい

✓ 共起の回数が多いればより可能性が高い

✓ いろいろな場所で共起すればより可能性が高い

✓ 人が少ない場所で共起すればより可能性が高い

# Social Strength

$$s_{ij} = \alpha D_{ij} + \beta F_{ij} + \gamma$$

ユーザ  $i$  と  $j$  の social strength

- ✓ 大きいほど友達っぽい
- ✓ これを推定したい

$i$  と  $j$  の diversity (後述)

- ✓ いろいろな場所で共起しているかを評価

$i$  と  $j$  の weighted frequency (後述)

- ✓ 人が少ない場所で共起しているかを評価

$D, F$  を説明変数、 $s$  を応答変数とする線形回帰で  
係数  $\alpha, \beta, \gamma$  を求める

# Diversity $D$

ユーザ  $i$  と  $j$  がいろいろな場所で共起しているかどうかを評価

例)

- ✓ 友達！ → 東京駅で1回、つくば駅で1回、大宮駅で1回共起した二人
- ✓ 友達？ → 東京駅で3回共起した二人

エントロピーを用いて評価

- ✓ 共起した場所のベクトルのエントロピーが高いほど友達っぽい
- ✓ 例)
  - ✓  $(1, 1, 1)$  -> エントロピー大
  - ✓  $(3, 0, 0)$  -> エントロピー小

# Weighted Frequency $F$

ユーザ  $i$  と  $j$  が人の少ない場所で共起しているかどうかを評価

例)

- ✓ 友達！ → 東京駅のマックと渋谷駅のマックで共起した二人
- ✓ 友達？ → 東京駅と渋谷駅で共起した二人

場所のエントロピーを用いて評価

- ✓ 多くの人がチェックインする場所はエントロピーが大きい
- ✓  $F = \sum_k (k \text{ での共起回数}) \times (\text{場所 } k \text{ のエントロピー})$

# Online Search of Overlapping Communities

13/09/07

SIGMOD勉強会 - Yuto Yamaguchi

# Problem

✓ クエリノードv0が与えられた時、v0を含むコミュニティを全て返す

## ✓ Note

- ✓ コミュニティの全列挙ではない
- ✓ オンラインクエリ
  - ✓ すぐ結果を返す
- ✓ コミュニティの重複を許す

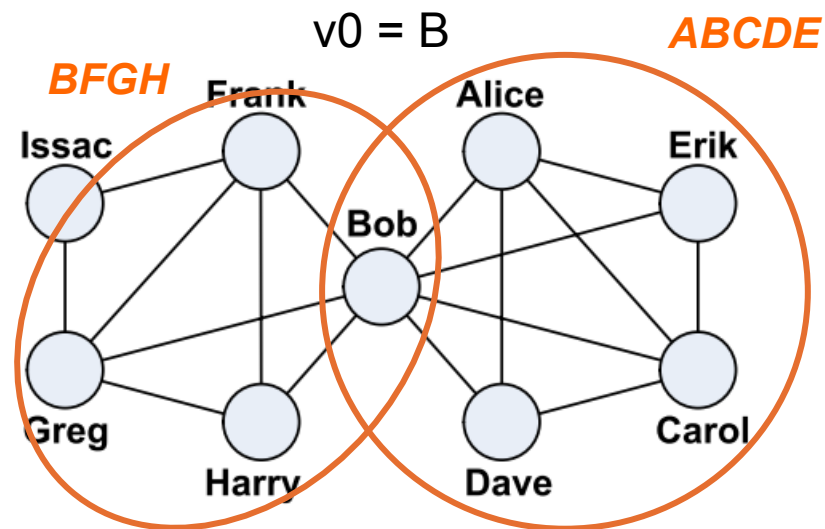


Figure 1: A toy social graph

# $\gamma$ -quasi-k-clique

以下を満たすサブグラフ

✓ ノード数が  $k$

✓ (密度)  $\geq \gamma \frac{k(k-1)}{2}$  完全グラフの  
密度

✓  $0 \leq \gamma \leq 1$

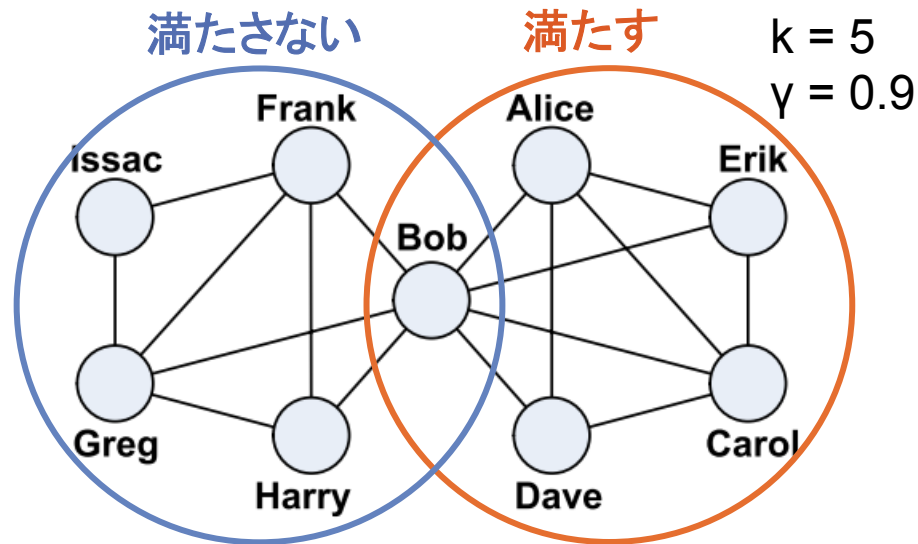


Figure 1: A toy social graph



# Clique Graph

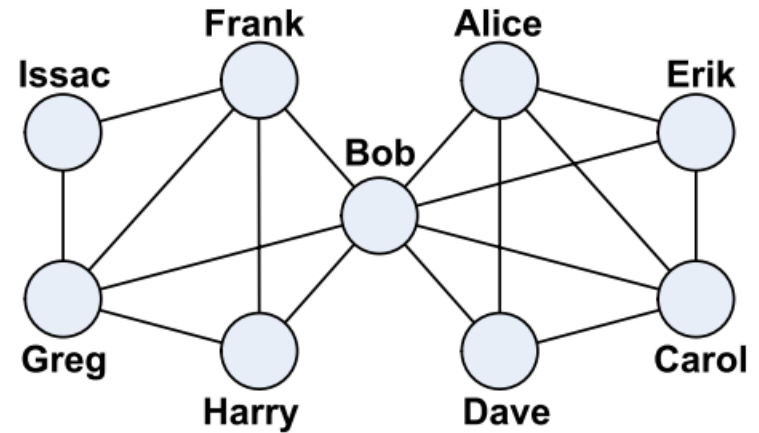
元のグラフから抽出されたクリークを  
ノードとするグラフ

✓  $\alpha$ -adjacency

- ✓ 二つの  $\gamma$ -quasi- $k$ -clique ( $k$ は同じ) は  
少なくとも  $\alpha$  個のノードを共有する時隣接

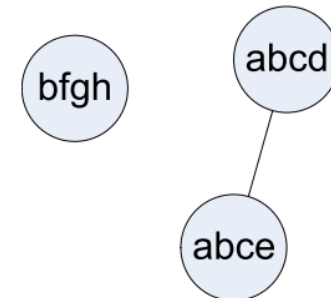
✓ **Clique component**

- ✓ Clique graphにおける連結成分に含まれる  
クリークの和集合



Original graph

$k = 4$   
 $\gamma = 1$   
 $\alpha = 3$



Clique graph

# $(\alpha, \gamma)$ -OCS

OCS: Overlapping Community Search

グラフG

パラメータ $k, \alpha, \gamma$

クエリノード $v_0$



が与えられた時

$v_0$ を含む全てのclique componentを見つける

# Exact Algorithm

- ✓ `next_clique()`
  - ✓ `v0`を含む、まだ訪れていない $\gamma$ -quasi-k-cliqueを返す
- ✓ `expand()`
  - ✓ 与えた $\gamma$ -quasi-k-cliqueを含む最大のclique componentを返す

## 1. `next_clique(v0 = B)`

1. -> ABCE

## 2. `expand(ABCE)`

1. -> ABCDE

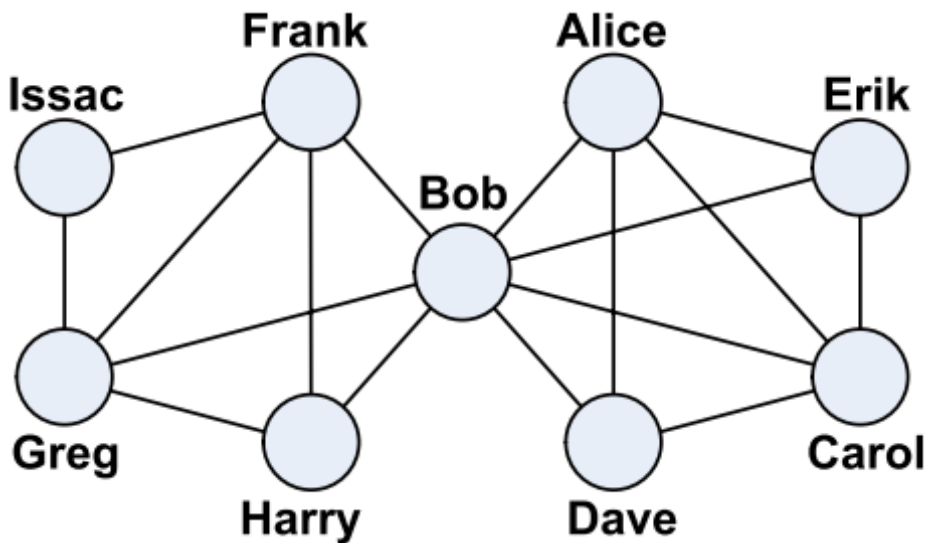
## 3. `next_clique(v0 = B)`

1. -> BFGH

## 4. `expand(BFGH)`

1. -> BFGH

## 5. 終了



詳細は論文... (近似アルゴリズムもあるよ!)