Mining Newsgroups Using Networks Arising From Social Behavior by Rakesh Agrawal et al.

Presented by Will Lee
wwlee1@uiuc.edu

September 28, 2004
Motivation

• IR on newsgroups is challenging due to lack of connection among documents
  – Unlike WWW, can not use PageRank to improve the retrieval performance

• An automatically-generated social network within a newsgroup may help IR and text mining applications
Methods Overview

• Classify authors as “for” or “against” a topic
Methods Overview

- Classify authors as “for” or “against” a topic
- Uses graph-theoretic approach to partition the interaction graph into two partitions
Methods Overview

- Classify authors as “for” or “against” a topic

- Uses graph-theoretic approach to partition the interaction graph into two partitions
  - graph nodes = users
Methods Overview

• Classify authors as “for” or “against” a topic

• Uses graph-theoretic approach to partition the interaction graph into two partitions
  – graph nodes = users
  – interaction (graph edges) = an user replying to another
Methods Overview

• Classify authors as “for” or “against” a topic

• Uses graph-theoretic approach to partition the interaction graph into two partitions
  
  – graph nodes = users
  
  – interaction (graph edges) = an user replying to another

• Assumptions
Methods Overview

• Classify authors as “for” or “against” a topic

• Uses graph-theoretic approach to partition the interaction graph into two partitions
  – graph nodes = users
  – interaction (graph edges) = an user replying to another

• Assumptions
  – New posts contain opposite comments against parent posts
Methods Overview

- Classify authors as “for” or “against” a topic

- Uses graph-theoretic approach to partition the interaction graph into two partitions
  - graph nodes = users
  - interaction (graph edges) = an user replying to another

- Assumptions
  - New posts contain opposite comments against parent posts
  - There are only two groups of users with roughly the same size
Graph Partitioning

- Define a graph $G(V, E)$
Graph Partitioning

- Define a graph $G(V, E)$
- $V = \text{newsgroup participants}$
Graph Partitioning

- Define a graph $G(V, E)$

- $V =$ newsgroup participants

- $e \in E$ where $e = (v_i, v_j)$ and $v_i, v_j \in V$ such that $v_i$ has responded to a post by $v_j$
Graph Partitioning

• Define a graph $G(V, E)$

• $V =$ newsgroup participants

• $e \in E$ where $e = (v_i, v_j)$ and $v_i, v_j \in V$ such that $v_i$ has responded to a post by $v_j$

• Goal is to find set of vertices $F$ (for) and $A$ (against)
Graph Partitioning

• Define a graph $G(V, E)$

• $V = \text{newsgroup participants}$

• $e \in E$ where $e = (v_i, v_j)$ and $v_i, v_j \in V$ such that $v_i$ has responded to a post by $v_j$

• Goal is to find set of vertices $F$ (for) and $A$ (against)

• Maximize the cut function $f(F, A) = |E \cap (F \times A)|$ (NP-complete problem)
Graph Partitioning

- Define a graph $G(V, E)$
- $V =$ newsgroup participants
- $e \in E$ where $e = (v_i, v_j)$ and $v_i, v_j \in V$ such that $v_i$ has responded to a post by $v_j$
- Goal is to find set of verticies $F$ (for) and $A$ (against)
- Maximize the cut function $f(F, A) = |E \cap (F \times A)|$ (NP-complete problem)
- Uses spectral partitioning for efficiency
Turning Social Behavior Into Graph Problem

For

Alice

Reply to

Cindy

Max Cut

Bob

Dan

Against

Elaine
Graph Partitioning Methods

1. EV Algorithm

(a) Co-citation matrix $D = GG^T$ with weighted edge $w = \#$ of people “co-cited” by author $u_1$ and $u_2$. Think of D as a similarity matrix for author $u_i$ and $u_j$.

(b) Second eigenvector of $D$ is a good approximation of $G$’s bipartition
Graph Partitioning Methods

1. EV Algorithm
   (a) Co-citation matrix $D = GG^T$ with weighted edge $w = \#$ of people “co-cited” by author $u_1$ and $u_2$. Think of $D$ as a similarity matrix for author $u_i$ and $u_j$.
   (b) Second eigenvector of $D$ is a good approximation of $G$’s bipartition

2. EV + KL
   (a) Uses the Kernighan-Lin heuristic to improve the partitioning
Graph Partitioning Methods

1. EV Algorithm

   (a) Co-citation matrix $D = GG^T$ with weighted edge $w = \#$ of people “co-cited” by author $u_1$ and $u_2$. Think of $D$ as a similarity matrix for author $u_i$ and $u_j$.

   (b) Second eigenvector of $D$ is a good approximation of $G$’s bipartition

2. EV + KL

   (a) Uses the Kernighan-Lin heuristic to improve the partitioning

3. EV (Constrained) and EV + KL (Constrained)

   (a) Identify some “for” and “against” authors, group them as one node
Graph Partitioning Methods

1. EV Algorithm
   
   (a) Co-citation matrix $D = GG^T$ with weighted edge $w = \#$ of people “co-cited” by author $u_1$ and $u_2$. Think of D as a similarity matrix for author $u_i$ and $u_j$.
   
   (b) Second eigenvector of $D$ is a good approximation of $G$’s bipartition

2. EV + KL
   
   (a) Uses the Kernighan-Lin heuristic to improve the partitioning

3. EV (Constrained) and EV + KL (Constrained)
   
   (a) Identify some “for” and “against” authors, group them as one node

4. Iterative Classification
Graph Partitioning Methods

1. EV Algorithm
   
   (a) Co-citation matrix $D = GG^T$ with weighted edge $w = \#$ of people “co-cited” by author $u_1$ and $u_2$. Think of $D$ as a similarity matrix for author $u_i$ and $u_j$.
   
   (b) Second eigenvector of $D$ is a good approximation of $G$’s bipartition

2. EV + KL
   
   (a) Uses the Kernighan-Lin heuristic to improve the partitioning

3. EV (Constrained) and EV + KL (Constrained)
   
   (a) Identify some “for” and “against” authors, group them as one node

4. Iterative Classification
   
   (a) Initialize: Label “for” and “against” for a small number of people in the newsgroup
(b) Iterate $m$ times:
(b) Iterate $m$ times:
   i. Calculate the $s(v_i)$ for each node $v_i$. The weight $w_{ij}$ is the weight between node $v_j$ and $v_i$:

   $$s(v_i) = \frac{\sum_j -s(v_j) \times w_{ij}}{\sum_j w_{ij}}$$
(b) Iterate \( m \) times:
   
i. Calculate the \( s(v_i) \) for each node \( v_i \). The weight \( w_{ij} \) is the weight between node \( v_j \) and \( v_i \): 

\[
    s(v_i) = \frac{\sum_j -s(v_j) \times w_{ij}}{\sum_j w_{ij}}
\]

   ii. Sort the labels (sign of \( s(v_i) \)) by confidence (|\( s(v_i) \)|)
(b) Iterate $m$ times:

i. Calculate the $s(v_i)$ for each node $v_i$. The weight $w_{ij}$ is the weight between node $v_j$ and $v_i$:

$$s(v_i) = \frac{\sum_j -s(v_j) \times w_{ij}}{\sum_j w_{ij}}$$

ii. Sort the labels (sign of $s(v_i)$) by confidence ($|s(v_i)|$)

iii. Accept $k = N \times \frac{i}{m}$ labels where $i = \text{iteration}$, $m = \text{total iterations}$, and $N = \text{number of instances in test data}$
Evaluation

- Uses three newsgroups – Abortion, Gun Control, and Immigration

- Manually tag 50 random people in the “for” or “against” categories

- Comparing with classic classification algorithms (Naive Bayes & SVM) that work on message content

<table>
<thead>
<tr>
<th></th>
<th>Abortion</th>
<th>Gun Control</th>
<th>Immigration</th>
</tr>
</thead>
<tbody>
<tr>
<td>Majority</td>
<td>57%</td>
<td>72%</td>
<td>54%</td>
</tr>
<tr>
<td>SVM</td>
<td>55%</td>
<td>42%</td>
<td>55%</td>
</tr>
<tr>
<td>Naive Bayes</td>
<td>50%</td>
<td>72%</td>
<td>54%</td>
</tr>
<tr>
<td>Iterative</td>
<td>67%</td>
<td>80%</td>
<td>83%</td>
</tr>
<tr>
<td>EV/EV+KL</td>
<td>73%/75%</td>
<td>78%/74%</td>
<td>50%/52%</td>
</tr>
<tr>
<td>Constrained EV/EV+KL</td>
<td>73%/73%</td>
<td>84%/82%</td>
<td>88%/88%</td>
</tr>
</tbody>
</table>

- Also, sensitivity experiments show more posts = more bias posts = higher accuracy
Contributions / Limitations

• Contributions
  – Apply graph-theoretic algorithms to a new domain
  – Sensitivity analysis on simulated newsgroup data

• Limitations
  – Assume users post against each other, may not be true in some newsgroups (technical ones)
  – Constrained and iterative method still need training data
  – Should justify why the constrained methods perform much better than the unconstrained ones
Discussion Questions

• How does user partitioning help IR?

• In a complex web of discussions within a newsgroup, users may not belong to the same “for” or “against” group for all topics. How can this system be applied on such newsgroup?

• How is this system similar to the PageRank algorithm? Is there any other way to draw connection among the newsgroup postings?