Do more Views of a Graph help?
Community Detection and Clustering in Multi-Graphs

Evangelos E. Papalexakis\textsuperscript{1}, Leman Akoglu\textsuperscript{2},
Dino Ienco\textsuperscript{3}
\textsuperscript{1}Carnegie Mellon University, \textsuperscript{2}Stony Brook University,
\textsuperscript{3}IRSTEA, France

International Conference on Data Fusion 2013
Roadmap

• Introduction
• Method Description
• Experimental Evaluation
• Data Mining Case Study
• Conclusions
Consider a network of authors, e.g. DBLP

Multiple possible views of the network: e.g.
1) who-cites-whom,
2) co-authorship,
3) using same words on title

We have a Multi-view Graph!
Motivation: Another Toy Example

Fig. 1. (a) Example multi-dimensional graph. Each different dimension (i.e. edge-type) is represented as a matrix: solid (-) (b), dashed (- -) (c) and dotted (..) edges (d).
Introduction

- We ask 2 fundamental questions:
  - **Q1**: Are these multiple use helpful in discovering communities? Are we better off just aggregating?
  - **Q2**: Is *any* extra view of the network useful? Can it also be harmful?
Introduction

• In this work
  ✷ We introduce 2 algorithms
    ▪ **MultiClus**: MDL based, works only for unweighted graphs
    ▪ **GraphFuse**: Tensor based, works for weighted graphs
  ✷ Experiments on real & synthetic data
  ✷ Data Mining Case Study on Real Data
Roadmap

- Introduction
- **Method Description**
- Experimental Evaluation
- Data Mining Case Study
- Conclusions
MultiClus Method

• Basic idea:
  ✦ Use Minimum Description Length (MDL)
  ✦ Penalize the number of blocks discovered throughout the different views
  ✦ Penalize the number of clusters
Our Objective Function: Total Encoding Cost

![Diagram of MultiClus Method]

\[ MDL_{objFunc} = \log^* n + \log^* k - \sum_{i=1}^{k} r_i \log_2 \left( \frac{r_i}{n} \right) + \sum_{l=1}^{m} \sum_{i=1}^{k} \sum_{j=1}^{k} \log^* n_1(B_{ij}^l) + E(B_{ij}^l) \]

- Code length for nodes
- Code length for clusters
- Cost of describing cluster assignments
- Cost of describing the ‘1’ of a block of the graph

NP-Hard! We use heuristic iterative algo based on top-down clustering
GraphFuse Method

• The multi-view Graph is a node X node X view tensor
  ✤ E.g. DBLP: author X author X relation
**GraphFuse Method**

\( X = \sum_{k=1}^{K} \alpha_k \beta_1 \gamma_1 + \cdots + \sum_{k=1}^{K} \alpha_k \beta_k \gamma_k \)

- \(a, b\): Give clustering assignment to communities
- \(c\): Gives the “participation” of each view of the graph to each community

Sparse PARAFAC decomposition
Roadmap

• Introduction
• Method Description
• Experimental Evaluation
• Data Mining Case Study
• Conclusions
Experiments

• We compare against state of the art algorithms that *ignore* the different views

• Aggregate across all different views and create a single graph

• Baselines:
  1. K-way spectral clustering
  2. Sum of spectral kernels k-means

• We measure NMI (Normalized Mutual Information) between output & ground truth labels
spectral clustering is based on the \textit{kzmeans} algorithm that is a \textit{kzway} spectral clustering over this aggregate \cite{31}.

Fig. 3. Spy-plots of 3 views in DBLP-1

Fig. 4. Spy-plots of 3 views in DBLP-2
Algorithm 1

**Input:**
- Multizgraph $G$ in tensor form $X$ of size $I \times J \times K$
- Number of clusters $R$
- Sparsity penalty factor $\lambda$

**Output:**
- Assignments to clusters $\alpha_{I}$ and $\alpha_{J}$
- Matrix $C$ of size $K \times R$ that shows the contribution of each one of the $K$ views to each one of the $R$ clusters

```plaintext
2. \{A, B, C\} - PRFC SLF $X^{R-1} + \lambda v f_i$
3. for $i = 1 \cdots I$
do
4. if $A(i,:)$ = 0
5. $\alpha_{I}(i) = R$
6. else
7. $\alpha_{I}(i) = \text{argmax} A(i,:)$
8. end if
9. end for
0. Repeat iteration 3-9 for all $J$ rows of $B$
```

Labels are output in $\alpha_{J}$

**Fig. 3** Utopv SYNTHETIC-3 and ubottomv SYNTHETIC-4 share the same clustering scheme, with different amount of cross edges and cluster densities. DIF multi-graph, by construction, is harder to cluster than SIM.

**Fig. 4** Spyzplots of 3 views in $D = LP$.

**Fig. 5** Spyzplots of 3 views in $D = LP$.

**B. Clustering accuracy**

In order to evaluate the performance of our proposed method, we use the Normalized Mutual Information, a widely used metric for computing clustering accuracy of a method against the desired ground truth clustering [23]. Moreover, we compare our methods in terms of NMI with two baseline approaches, which we briefly describe in the sequel.

**SELINE** algorithm sums all the adjacency matrices of a multizgraph obtaining a new aggregate sum matrix and applies a $k$-way spectral clustering over this aggregate [31]. The $k$-way spectral clustering is based on the $k$-means algorithm that is applied on the Laplacian of the sum matrix.

**SELINE** algorithm first constructs the spectral kernel for each graph view and then sums the spectral kernels summarizing all the dimensions of the multizgraph. Successively, the $k$-means algorithm is applied to the matrix containing the sum of the kernels in order to obtain the final clustering. Details for this algorithm may be found in [20].

In Table I we show the NMI results on all datasets for all methods. We observe that MULTICLUS always outperforms baseline methods on all synthetic datasets. For $G$RPH, it has good performance over SIM and SYNTHETIC-3 while for SYNTHETIC-4 the results are on par with the baselines. Recall that by construction SYNTHETIC-4 is difficult to cluster, hence the drop in performance for all methods.

With respect to the real datasets, G$RPH$ obtains the best scores over both $D = LP$ and $D = LP$, while MULTICLUS has comparable behaviour with the baselines. We notice that NMI scores are overall lower on real datasets as they have much less structure than the synthetic ones, in addition to a lot more noise. Nevertheless, G$RPH$ achieves significantly better accuracy compared to other methods. These encouraging results underline the merits of modeling the multizgraph clustering problem using tensors, as they seem to well exploit the interrelations of the views.
Results

<table>
<thead>
<tr>
<th>Dataset</th>
<th>BASELINE-1</th>
<th>BASELINE-2</th>
<th>MultiCLUS</th>
<th>GraphFuse-1</th>
<th>GraphFuse-2</th>
</tr>
</thead>
<tbody>
<tr>
<td>SYNTHETIC-1</td>
<td>0.77 ± 0.11</td>
<td>0.96 ± 0.06</td>
<td>1 ± 0</td>
<td>1 ± 0</td>
<td>1 ± 0</td>
</tr>
<tr>
<td>SYNT-2-sim</td>
<td>0.68 ± 0.12</td>
<td>0.97 ± 0.11</td>
<td>1 ± 0</td>
<td>1 ± 0</td>
<td>1 ± 0</td>
</tr>
<tr>
<td>SYNT-3-dif</td>
<td>0.54 ± 0.01</td>
<td>0.56 ± 0.02</td>
<td>0.90 ± 0.01</td>
<td>0.51 ± 0.17</td>
<td>0.67 ± 0.12</td>
</tr>
<tr>
<td>DBLP-1</td>
<td>0.12 ± 0.00</td>
<td>0.08 ± 0.01</td>
<td>0.11 ± 0.01</td>
<td>0.30 ± 0.02</td>
<td>0.29 ± 0.02</td>
</tr>
<tr>
<td>DBLP-2</td>
<td>0.08 ± 0.01</td>
<td>0.04 ± 0.00</td>
<td>0.04 ± 0.00</td>
<td>0.12 ± 0.02</td>
<td>0.09 ± 0.02</td>
</tr>
</tbody>
</table>

**TABLE I.** NMI clustering accuracy of proposed methods and competitors on all datasets. Our proposed methods achieve superior performance for all clustering tasks.

- Both proposed methods outperform the baselines
- For DBLP labels are manual, might not reflect the true community structure
Q2: Do More Views Always Help?

- Conducted exhaustive experiments
- Answer:
  - More views help **on average**
  - Might have one bad view that if take into account, decreases the quality of results
Roadmap

• Introduction
• Method Description
• Experimental Evaluation
• Data Mining Case Study
• Conclusions
Data Mining Case Study

• “RealityMining” dataset
  ✷ From MIT Media Lab
  ✷ Records “CALL”, “SMS”, “DEVICE” (Bluetooth), and “FRIEND” relations between students/faculty

• We applied GraphFuse
Data Mining Case Study

Fig. 6. Results on the four views of the REALITYMINING multi-graph. Red dashed lines outline the clustering found by GRAPHFUSE.

Fig. 5. View-by-cluster matrix $C$ by GRAPHFUSE; which essentially encodes the intensity of influence of each of the $K$ views on each of the $R$ clusters. The density of each view is clearly reflected here, but as an interesting future direction, we could potentially identify low quality views by observing how influential they are, according to $C$. 
Roadmap

- Introduction
- Method Description
- Experimental Evaluation
- Data Mining Case Study
- Conclusions
Conclusions

• We introduced 2 methods for Multi-view Graph Clustering
  - MultiClus, MDL based, unweighted
  - GraphFuse, Tensor based, weighted

• Outperform baselines that ignore Multi-view structure

• GraphFuse is able to tell which views are more helpful