Multilayer network clustering via graph embedding

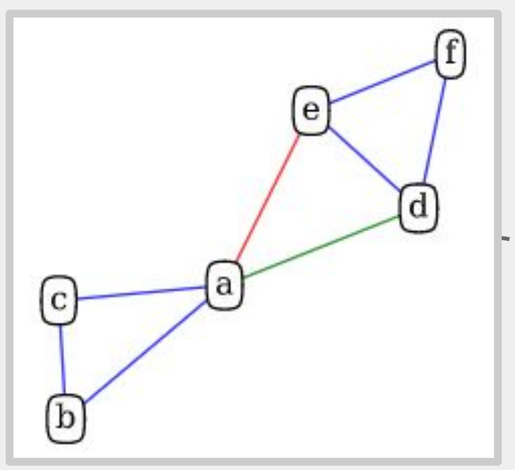
Yegor Piven yegor.piven.6168@student.uu.se
Niclas Samuelsson niclas.samuelsson.0401@student.uu.se
Daniel Strandberg daniel.strandberg.9338@student.uu.se

Supervisor: Matteo Magnani
Department of Information Technology
Uppsala University

Goal

Implement and evaluate embedding techniques for the problem of multilayer network clustering.

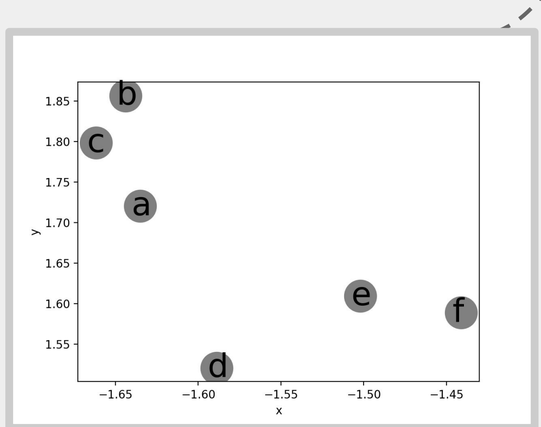
Network embedding



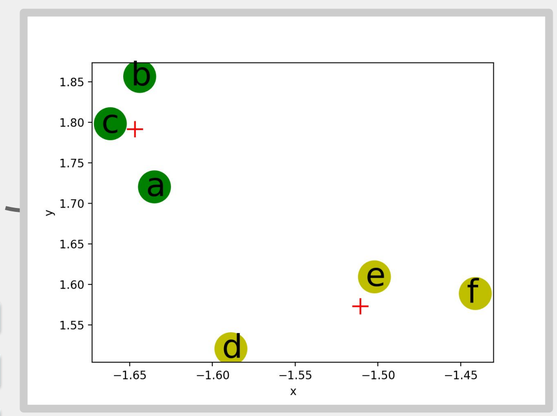
Take **biased random walks** through the network to get "text of nodes"

a -> d -> f -> d
d -> e -> a -> b
c -> b -> c -> a
…
…
f -> d -> e -> a
b -> a -> c -> b
d -> f -> d -> a

Embed this "text of nodes" into a vector space, using a NLP technique called *word2vec*



To **find the clusters**, we use KMeans on the embedding



Sidenote

Embeddings can be useful in a number of other applications, for example in link prediction!

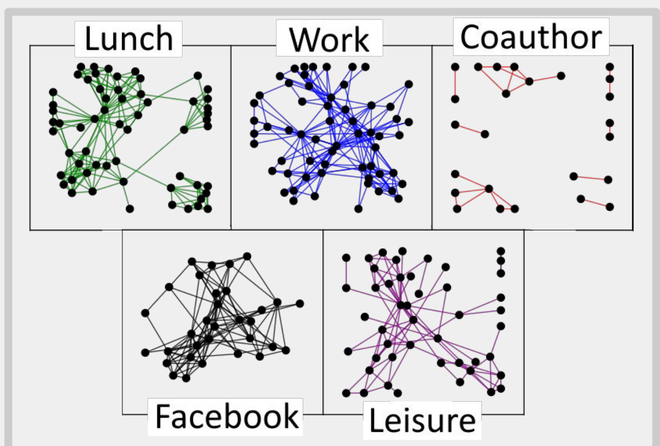
Handle multiple layers

Problem is finding the "text of nodes" for a network with more than one layer. Three ways suggested in [1]:

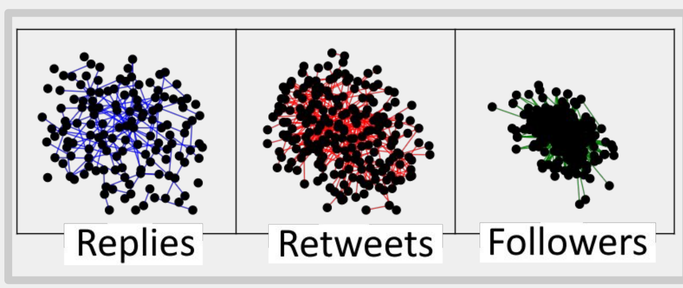
- **Layer aggregation:** Flatten the network into a 1D network (possibly weighted) and sample in the normal way.
- **Result aggregation**: View each layer as a different network and embed it on its own. Then append the vectors to each other in order to get the full embedding.
- **Layer co-analysis**: Sample paths in the network with a probability of jumping between layers.

Test data

Two networks with existing "ground truth" clusterings.



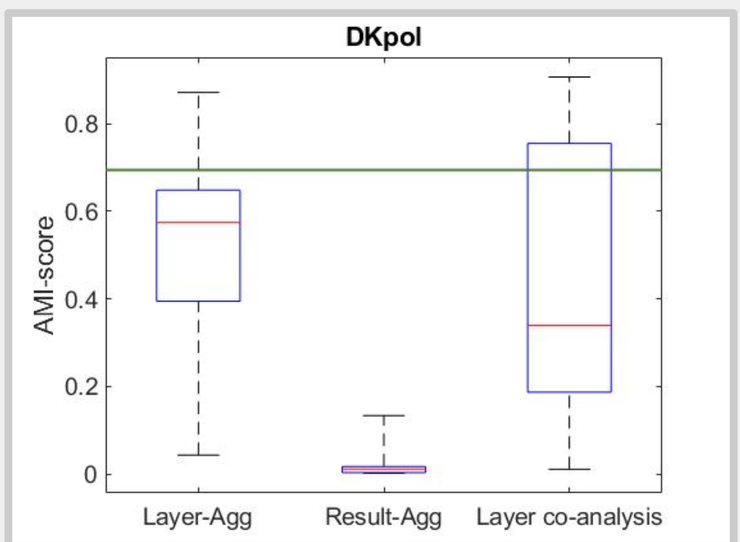
AUCS (61 actors):
Mutual interactions between some academics. The goal is to find the known research groups.

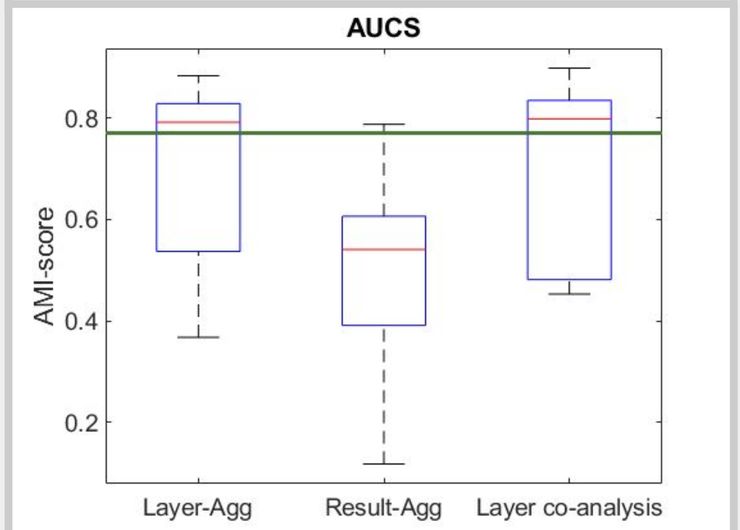


DKpol (491 actors):
Danish politicians' interactions on Twitter. The goal is to find the known political parties.

Results

We evaluate the results by measuring the **adjusted mutual information (AMI)** between the acquired clusterings and the given "ground truth". A value of 0 implies sharing no information, and 1 suggests identical clusterings, so a **higher score is better**.





Performance for different sets of hyperparameters, compared to generalized Louvain method (green line).

Layer aggregation appears to be most robust under different sets of hyperparameters.

Result aggregation generally presents worse results.

Layer-co analysis shows the most promise, but is sensitive to the tuning.

Performance is worse for larger network:
Are more walks needed?
Or structural reasons?

Conclusion

The embedding approach is promising, but is fairly sensitive to the tuning of hyperparameters. A well-tuned algorithm could achieve better results than the benchmark algorithm glouvain2, but tuning the embedding algorithm for clustering in an unsupervised setting is an open problem.

[1] Chen et al "Principled Multilayer Network Embedding" 15 September 2017