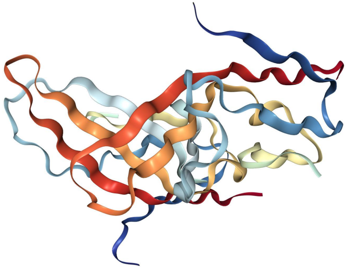
Neural Blockmodeling for Multilayer Networks

Marcin Pietrasik
Marek Reformat

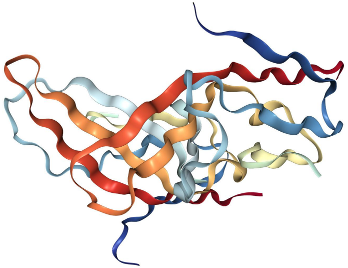
Outline

- Motivation and problem formulation
- Existing approaches
- Our proposed approach
- Evaluation procedure
- Results

Networks are everywhere!



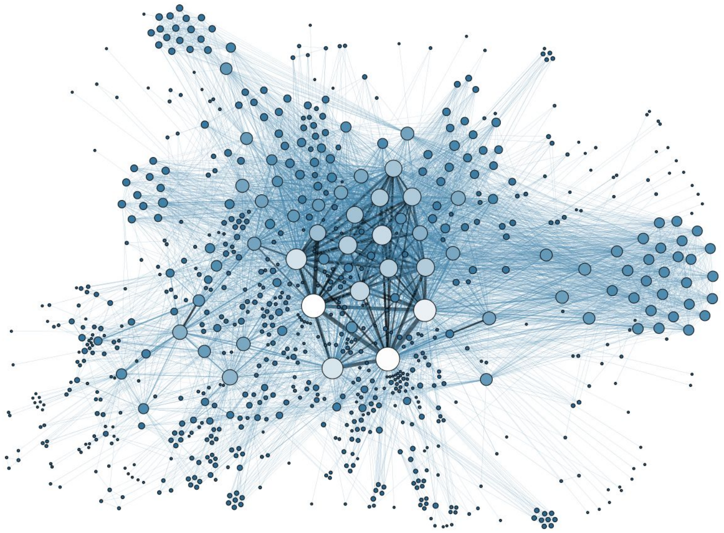
Networks are everywhere!



We need to find ways of accurately representing them

What are networks?

- Structures which capture the relations between discrete objects
- In a network, nodes (vertices) are connected together by links (edges).



Simple network example

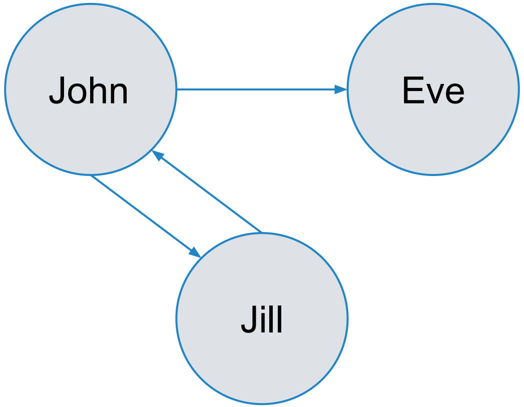
Textual

John is friends with Jill

Jill is friends with John

John is friends with Eve

Graphical

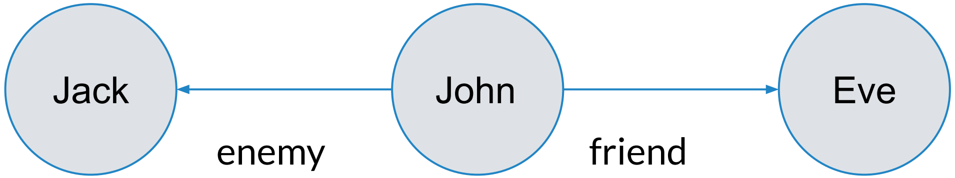


Adjacency Matrix

	John	Jill	Eve
John	0	1	1
Jill	1	0	0
Eve	0	0	0

Multilayer networks

- In a multilayer network, nodes are linked together by different types of relations
  - Ex. friend and enemy



Simple multilayer network example

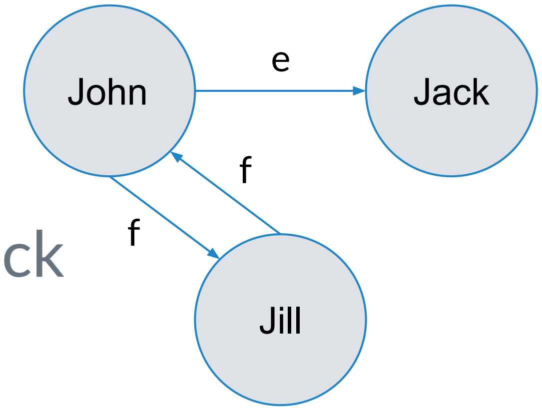
Textual

John is friends with Jill

Jill is friends with John

John is enemies with Jack

Graphical



Adjacency Matrix

Friend	John	Jill	Jack
John	0	1	0
Jill	1	0	0
Jack	0	0	0

Enemy	John	Jill	Jack
John	0	0	1
Jill	0	0	0
Jack	0	0	0

Problem

- Find a representation of a multilayer network that allows computers to reason with it in an intelligent way

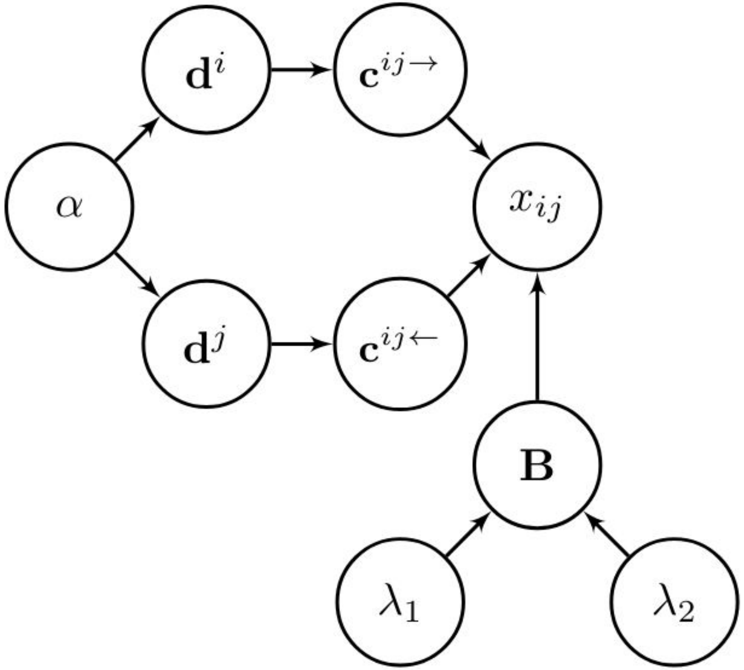
Existing approaches to network modeling

- Two common approaches are:
  - Blockmodeling
  - Embedding methods

Blockmodels

- Blockmodels decompose a network into probability distributions
- Each probability distribution represents a structural component of the network
- When sampled together, the probability distributions generate the network
- Usually rely on finding communities in network

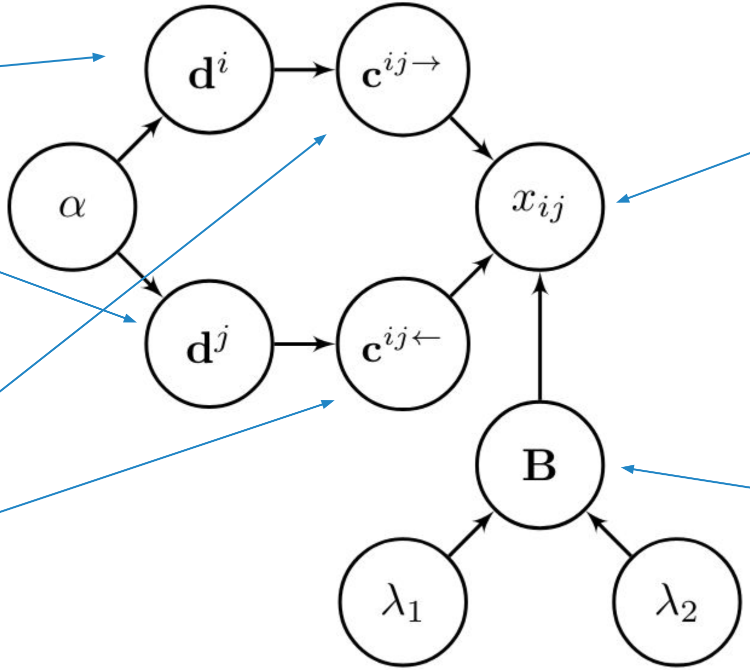
Blockmodels



Blockmodels

Community membership

Value of interaction between node *i* to node *j*



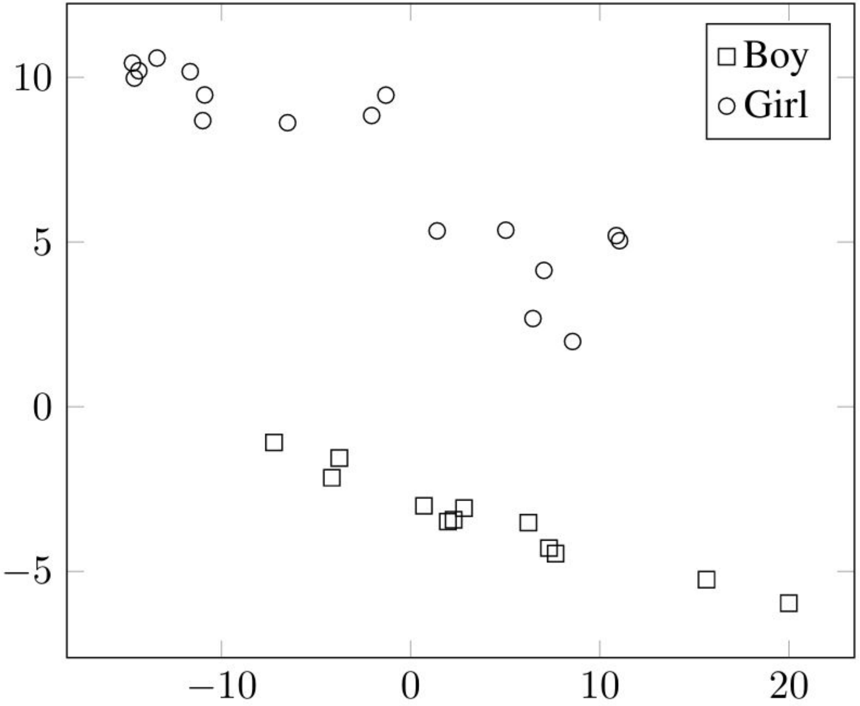
Community indicator

Community interactions

Embedding methods

- Use deep architectures to learn latent representations of nodes in an embedding space
- Usually rely on sampling paths in the network and embedding nodes which co-occur more often in these paths close to one another in the embedding space

Embedding methods



Drawbacks of existing approaches

- Blockmodels
  - Difficult to model nodes jointly and learn deep representations
  - Inference scheme oftentimes complicated
- Embeddings methods
  - Limited work in multilayer networks
  - Need to be used in conjunction with other methods to solve certain tasks

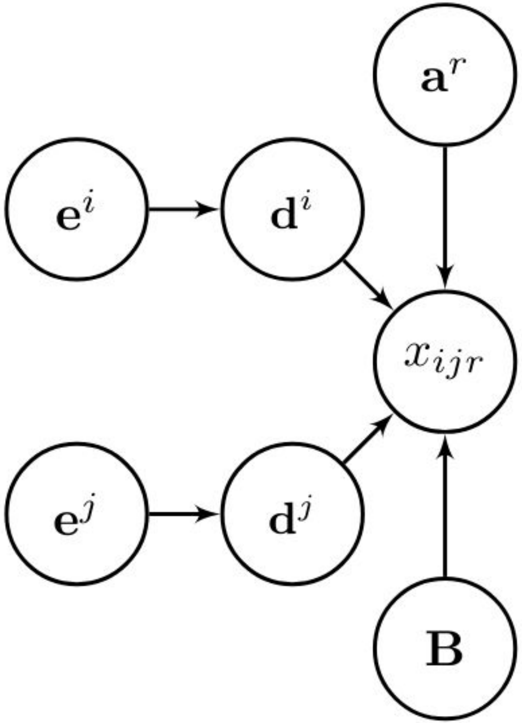
Our approach

- Our model fuses blockmodeling with embedding methods
  - Overcomes drawbacks of both
- Learns network communities as well as node embeddings
- Solved using a neural approach

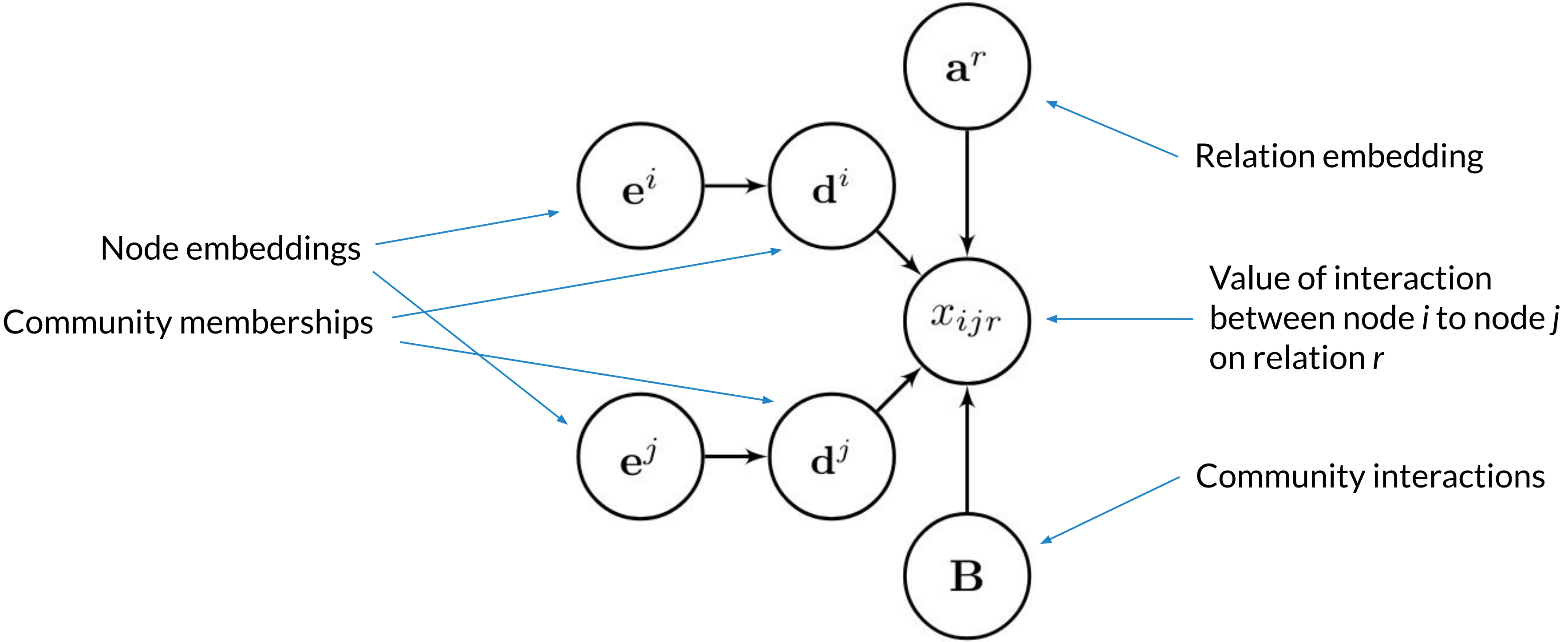
Our generative approach (high level)

1. Learn embedding for each node
2. Assign each node to a community using embedding
3. Learn interactions between communities
4. Extend interactions between communities to their constituent nodes
5. Learn and apply relational modifier to node interactions

Our generative approach (graphical)



Our generative approach (graphical)

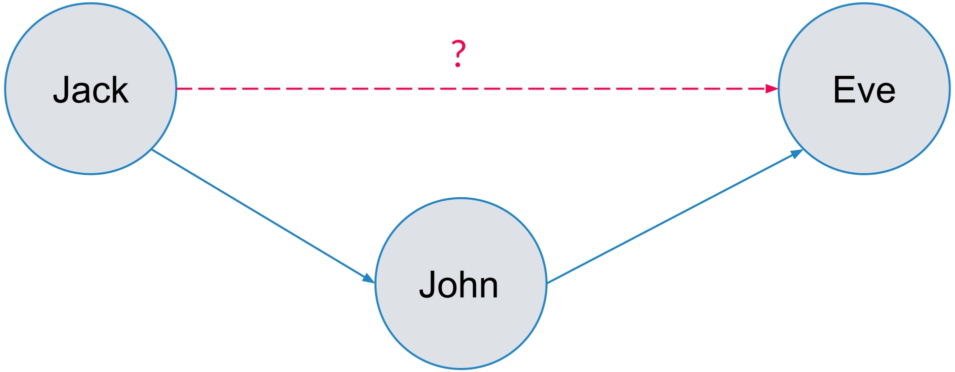


Evaluation

- Three tasks which are commonly used to evaluate performance of model:
  - Link prediction
  - Node classification
  - Community detection

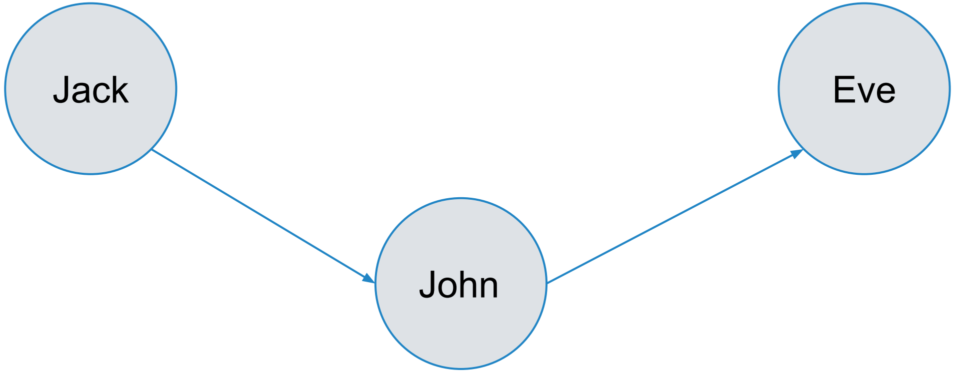
Link prediction

- Infer whether a link exists between two nodes



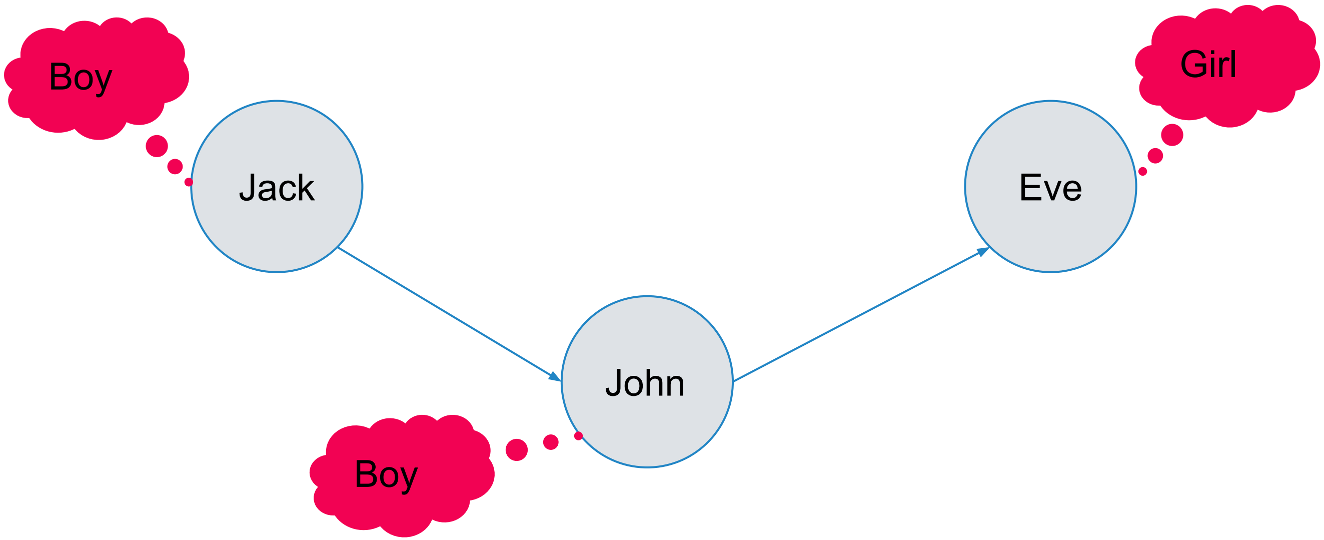
Node classification

- Assign a discrete label to a node



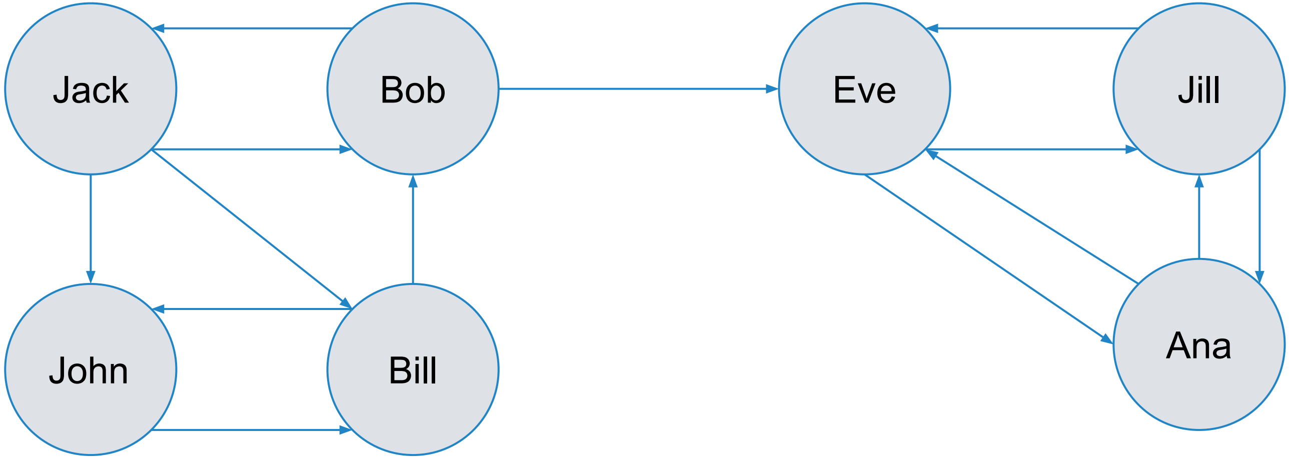
Node classification

- Assign a discrete label to a node



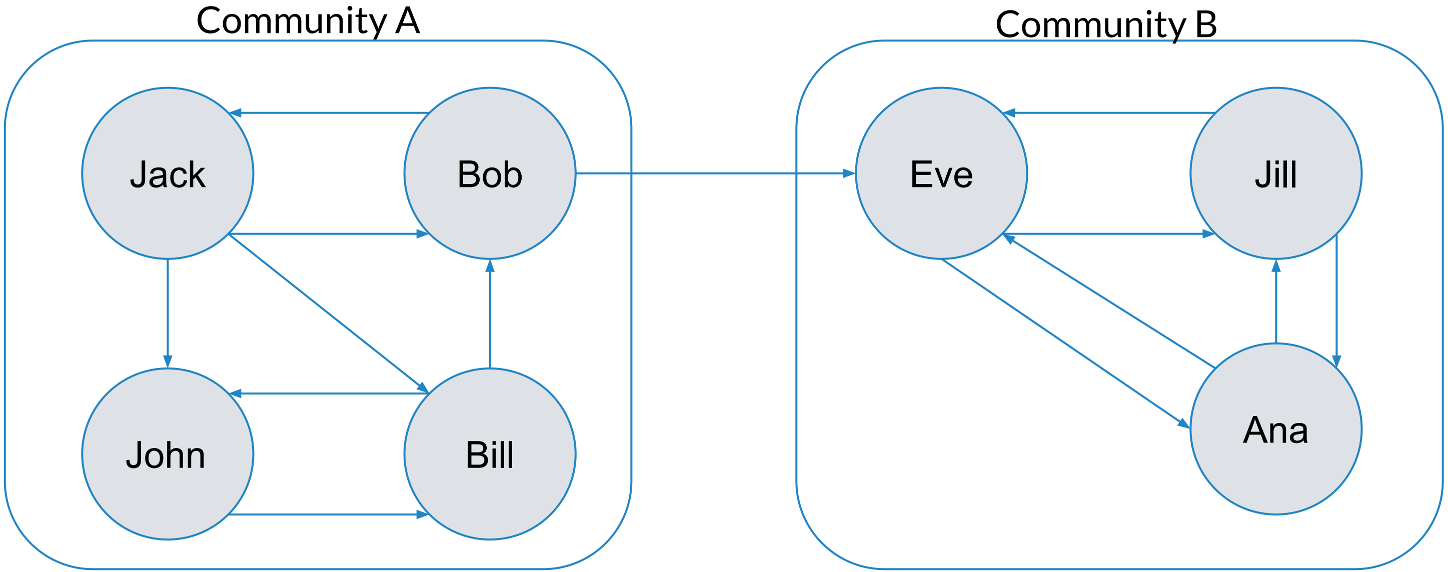
Community detection

- Detect communities (nodes with similar properties) in a network



Community detection

- Detect communities (nodes with similar properties) in a network



Datasets

- Trade: trading relations of various commodities between countries
- Vickers-Chan: social relations between students in Australian classroom
- Lazega: social relations between American lawyers
- Krebs: social relations between IT employees
- Twitter: interaction between Twitter users

Results: link prediction

TABLE II
LINK PREDICTION AUC SCORES (MEAN $\pm$ STANDARD DEVIATION) ON VARIOUS DATASETS

Method	Dataset				
	Trade	Vickers-Chan	Lazega	Krebs	Twitter
Blockmodels					
MMSB	0.8679 ± 0.0418	0.8153 ± 0.0420	0.8202 ± 0.0246	0.8335 ± 0.0759	0.7752 ± 0.0825
dMMSB	0.8768 ± 0.0102	0.8513 ± 0.0171	0.8155 ± 0.0040	0.8401 ± 0.0271	0.9166 ± 0.0023
fcMMSB	0.7746 ± 0.0422	0.7926 ± 0.0390	0.7642 ± 0.0246	0.8092 ± 0.0135	0.9030 ± 0.0033
DDBN	0.8525 ± 0.0145	0.8924 ± 0.0127	0.8386 ± 0.0056	0.9276 ± 0.0048	0.8589 ± 0.0040
Embeddings					
DeepWalk	0.5782 ± 0.0185	0.8340 ± 0.0183	0.7978 ± 0.0048	0.8269 ± 0.0099	0.6027 ± 0.0016
node2vec	0.5377 ± 0.0295	0.8214 ± 0.0203	0.7821 ± 0.0052	0.8183 ± 0.0079	0.6015 ± 0.0022
PMNE	0.6207 ± 0.0295	0.8457 ± 0.0138	0.8142 ± 0.0084	0.8887 ± 0.0052	0.7081 ± 0.0027
MNE	0.5444 ± 0.0696	0.8707 ± 0.0103	0.8246 ± 0.0078	0.8544 ± 0.0050	0.6001 ± 0.0036
MNB	0.8797 ± 0.0130	0.8707 ± 0.0172	0.8527 ± 0.0028	0.8948 ± 0.0071	0.8980 ± 0.0064

Our method

Results: node classification

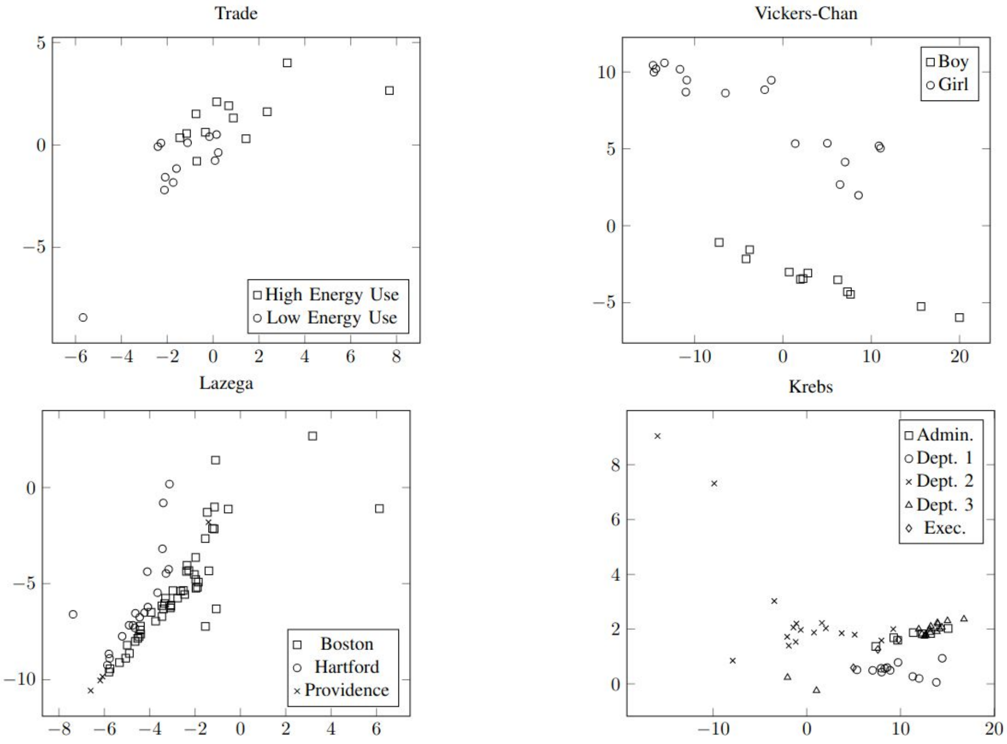
TABLE III

NODE CLASSIFICATION ACCURACY SCORES (MEAN $\pm$ STANDARD DEVIATION) ON VARIOUS DATASETS

Method	Dataset: Trade	Vickers-Chan	Lazega	Krebs
Deepwalk				
$E = 2$	0.3200 ± 0.1600	1.0000 ± 0.0000	0.7733 ± 0.0998	0.6833 ± 0.1856
$E = 10$	0.3200 ± 0.0980	1.0000 ± 0.0000	0.9333 ± 0.0596	0.8667 ± 0.1130
node2vec				
$E = 2$	0.3200 ± 0.0980	1.0000 ± 0.0000	0.7067 ± 0.1236	0.7167 ± 0.0850
$E = 10$	0.3200 ± 0.0980	1.0000 ± 0.0000	0.9200 ± 0.0267	0.8333 ± 0.0913
PMNE				
$E = 2$	0.3200 ± 0.0980	0.9000 ± 0.0816	0.9333 ± 0.0422	0.7333 ± 0.0133
$E = 10$	0.3200 ± 0.0980	0.9667 ± 0.0667	0.9200 ± 0.0499	0.8500 ± 0.0333
MNE				
$E = 2$	0.3800 ± 0.2088	1.0000 ± 0.0000	0.9333 ± 0.0667	0.7167 ± 0.1546
$E = 10$	0.3600 ± 0.1200	1.0000 ± 0.0000	0.9267 ± 0.0554	0.8500 ± 0.1041
MNB				
$E = 2$	0.7200 ± 0.1600	1.0000 ± 0.0000	0.7867 ± 0.0778	0.7333 ± 0.0624
$E = 10$	0.8600 ± 0.2010	1.0000 ± 0.0000	0.9200 ± 0.0581	0.8083 ± 0.1057

Our method

Results: node classification



Results: community detection

TABLE IV
COMMUNITY DETECTION CONDUCTANCE (COND.) AND NORMALIZED CUT (NC) SCORES (MEAN ± STANDARD DEVIATION) ON VARIOUS DATASETS

	Dataset									
	Trade		Vickers-Chan		Lazega		Krebs		Twitter	
Method	Cond.	NC	Cond.	NC	Cond.	NC	Cond.	NC	Cond.	NC
MMSB										
$K=2$	1.6780 ±0.3111	2.5277 ±0.1982	1.0622 ±0.3411	1.6465 ±0.4282	1.0008 ±0.1518	1.7353 ±0.1933	1.3459 ±0.4599	1.9175 ±0.6069	1.2950 ±0.2486	2.1856 ±0.2131
$K=4$	2.0239 ±0.2510	2.5731 ±0.2517	1.2673 ±0.1925	1.6738 ±0.2397	1.3843 ±0.1235	1.8314 ±0.1614	1.4023 ±0.1989	1.8012 ±0.2607	1.6420 ±0.1751	2.1556 ±0.1907
dMMSB										
$K=2$	1.7184 ±0.2422	2.5604 ±0.1937	1.2788 ±0.3864	1.7502 ±0.1590	1.2291 ±0.1458	1.8821 ±0.1097	1.6324 ±0.3734	2.2407 ±0.2137	3.2061 ±1.9318	3.7244 ±1.5897
$K=4$	1.9284 ±0.1880	2.4654 ±0.1731	1.3704 ±0.2819	1.7691 ±0.3008	1.4409 ±0.0750	1.9122 ±0.0916	1.6047 ±0.1561	2.0690 ±0.1797	2.3241 ±0.9502	2.8486 ±0.9661
DDBN										
$K=2$	1.8312 ±0.0000	2.0096 ±0.0000	1.3398 ±0.0000	1.6351 ±0.0000	1.8124 ±0.1721	2.0123 ±0.0913	1.5638 ±0.0000	1.6492 ±0.0000	1.1670 ±0.1218	1.8951 ±0.1522
$K=4$	4.3723 ±0.2296	4.6180 ±0.2055	1.3142 ±0.1131	1.6433 ±0.1043	1.1887 ±0.0404	1.5504 ±0.0420	0.8777 ±0.2552	1.1465 ±0.2749	1.5977 ±0.2244	1.8010 ±0.3806
fcMMSB										
$G=2$	2.2501 ±0.9639	2.7666 ±0.7634	1.0812 ±0.2471	1.6067 ±0.2686	1.4995 ±0.2405	1.9915 ±0.0967	1.0538 ±0.2148	1.6878 ±0.2854	1.4734 ±1.3274	1.6958 ±1.2327
$G=4$	2.1971 ±0.5501	2.6536 ±0.4846	1.3488 ±0.2156	1.7595 ±0.2649	1.3570 ±0.1093	1.7906 ±0.1353	1.2889 ±0.1891	1.6787 ±0.2336	0.9198 ±0.2550	1.1103 ±0.2310
MNB										
$K=2$	1.4900 ±0.0000	2.5698 ±0.0000	0.7606 ±0.0000	1.3161 ±0.0000	1.1047 ±0.0330	1.7754 ±0.0440	0.4386 ±0.0000	0.6692 ±0.0000	2.8168 ±0.2837	3.0646 ±0.3090
$K=4$	1.7313 ±0.0289	2.2945 ±0.0269	1.1643 ±0.0367	1.5456 ±0.0381	1.3035 ±0.0815	1.7083 ±0.0974	0.8908 ±0.1480	1.1395 ±0.1941	3.4614 ±0.4343	3.7755 ±0.4530

Our method

Conclusion

- We introduced a method for modeling multilayer networks that fuses blockmodels with embedding methods in a neural framework
- Results show that our model is competitive with or better than state-of-the-art methods
- Code and datasets necessary to replicate our results may be found on GitHub at: https://github.com/mpietrasik/mnb

Attributions

"Amazon Prime Air Boeing 767-306(ER)(BCF)" (CC BY 2.0) by Patrick Feller

“CD83-protein-structure.jpg” (CC BY 4.0) by Machacdav

“Social_Network_Analysis_Visualization.png” (CC BY-SA 3.0) by Martin Grandjean

Thanks!

Questions can be sent to pietrasi@ualberta.ca