Link Prediction via Community Detection in Bipartite Multi-Layer Graphs (Supplementary Material)

Maksim Koptelov
Normandie Univ, UNICAEN, ENSICAEN, CNRS - UMR GREYC, 14000 Caen, France
maksim.koptelov@unicaen.fr

Albrecht Zimmermann
Normandie Univ, UNICAEN, ENSICAEN, CNRS - UMR GREYC, 14000 Caen, France
albrecht.zimmermann@unicaen.fr

Bruno Crémilleux
Normandie Univ, UNICAEN, ENSICAEN, CNRS - UMR GREYC, 14000 Caen, France
bruno.cremilleux@unicaen.fr

Lina Soualmia
Normandie Univ, UNIROUEN, ULH, INSAR - LITIS-TIBS, 76800 Rouen, France
lina.soualmia@chu-rouen.fr

ACM Reference Format:
Maksim Koptelov, Albrecht Zimmermann, Bruno Crémilleux, and Lina Soualmia. 2020. Link Prediction via Community Detection in Bipartite Multi-Layer Graphs (Supplementary Material). In *The 35th ACM/SIGAPP Symposium on Applied Computing (SAC '20), March 30-April 3, 2020, Brno, Czech Republic.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3341105.3373874

1 EXTENSIVE EVALUATION ON A BIGGER DATA SET

In this document we present description and results of additional experiments performed to test different aspects of our approach described in detail in "Link prediction via community detection in bipartite multi-layer graphs" paper. They include general performance and scalability on a bigger data set, and verification of predicted interactions with a confidence score.

1.1 Performance on the IUPHAR

The five benchmark data sets on which we have reported to far are relatively small and dense, as shown in Table 1 in the main article. In this section, we therefore contrast those results with those on the IUPHAR data set, having 6 layers. We have taken that data set from [3], where it is also described in detail. Note, that negative edges were not removed from IUPHAR imitating benchmark data sets structure. Instead, edge labels were ignored for measures computation step, what allowed us to make a comparison of the results with our former approach in [3]. The data set basic properties are also presented in Table 1 in the main part, which shows that IUPHAR is significantly larger.

Given that we have shown above that fixing parameter values internally gives effectively the same results as reporting the best result on the testing data, we did the latter for IUPHAR to save time. The results are shown in Table 4. In addition to averaged AUC, AUPR we also report standard deviation values when it is applicable. IUPHAR is too large to search the parameter space for m starting from $m = 1$ with step size 1 – instead we used step size 10 for the grid search and searched in a more fine-grained manner once we had found a maximum in this way. As Table 4 shows, using either spectral partitioning or Louvain clearly improves on the results of the baseline[1]. Contrary to the results on the benchmark data sets, however, spectral partitioning *outperforms* Louvain, even though the two are close for node-community matching. The table also shows a very surprising result in that spectral partitioning with node-community matching does best when not creating communities at all! In that case it is the matching technique that does the heavy lifting and effectively treats each entity as a community of size 1 when the time comes to predict new links.

To evaluate whether this is a phenomenon that is specific to IUPHAR or occurs more generally, we compare the results for $m = 0$ on the benchmark data to the ones achieved by parameter-optimized spectral partitioning in Table 5. As the table shows, optimizing parameters *does* provide a performance gain, sometimes strongly so, as in the case of NR. Yet at the same time, the results for $m = 0$ are acceptable. This indicates both that those benchmark data sets are not fully representative of the problem setting, and that for large

[1]Baseline results taken from [3], where AUPR was not reported.

SAC '20, March 30-April 3, 2020, Brno, Czech Republic

ACM ISBN 978-1-4503-6866-7/20/03…$15.00
https://doi.org/10.1145/3341105.3373874

Table 4: Performance ceiling on IUPHAR (* – for 1 fold in average; ** – Spectral partitioning without thresholding optimization)

Approach	Measure	Optimal parameters	Performance				Running time*, s
			AUC	σ	AUPR	σ	
Baseline	-	eta=0.2, beta=0.7	0.57	-	-	-	11.9x8137
Spectral partitioning	JC_{CC}	m=400, default**	0.74	0.02	0.01	0.00	3477.82
	JC_{NC}	m=0, w/o threshold	0.85	0.00	0.01	0.00	2484.91
Louvain algorithm	JC_{CC}	resolution=0.1	0.61	0.01	0.10	0.01	5263.97
	JC_{NC}	resolution=0.8	0.82	0.01	0.02	0.00	2702.62

Table 5: Link prediction without community detection vs spectral partitioning with optimal parameters and *NC* matching

Data set	Without communities (m=0)				Optimal m and threshold			
	AUC	σ	AUPR	σ	AUC	σ	AUPR	σ
Enzyme	0.88	0.01	0.11	0.01	0.92	0.01	0.29	0.05
GPCR	0.78	0.02	0.18	0.02	0.85	0.02	0.27	0.04
IC	0.85	0.01	0.25	0.02	0.89	0.02	0.43	0.06
NR	0.68	0.08	0.16	0.05	0.78	0.05	0.25	0.09
Kinase	0.85	0.01	0.32	0.03	0.86	0.01	0.36	0.03

data one could do some quick-shot prediction using $m = 0$ and node-community matching before going to the effort of optimizing parameters.

1.2 Verification of predicted drug-target interactions

In this experiment, we verify our approach with different settings by predicting new interactions and providing a confidence value for the prediction results. We compare again with the baseline method from [3]. We perform verification on the IUPHAR data. We use the full data, i.e. without splitting into train and test sets and predict edges that are not present in IUPHAR *at all*. We reuse the optimal found in the preceding section (an approach similar to an internal validation).

We search for occurrences of predicted results in external resources, introduce a confidence score, and, finally, give feedback. As an external resource we use the Unified Medical Language System (UMLS) [1], which is composed of three knowledge sources: the Metathesaurus, the Semantic Network, and the SPECIALIST Lexicon. We define verification as a multi-step process: 1) Map drugs and targets from predicted interactions with the Metathesaurus vocabulary. For that, we convert the *IUPHAR drug/target id* into the *IUPHAR drug/target provisional name*; 2) Retrieve semantic types for all predicted drugs and targets. For this step, we use the MetaMap tool [2]; 3) Match each predicted drug-target pair with retrieved semantic type relations using the Semantic Network; 4) For each matched relation compute a confidence score defined as:

$$score(d,t) = \frac{\sum_{i,j}^{N} \frac{conf(d_i)}{1000} \cdot \frac{conf(t_j)}{1000}}{N} \times N,$$

where $conf(d_i)$, $conf(t_j)$ denote confidence scores of a drug concept i and a target concept j respectively, and N is a number of matched semantic type relations. Finally, 5) Generate feedback for each predicted drug-target pair (see Table 7 for examples), and compute a summary for all predictions. Note that steps 1-2 can be precomputed for all drugs and targets in a data set.

We used Precision@20 to predict new interactions, and performed that for each drug in the set. As an evaluation criterion, we report the percentage of verified predictions in total, as well as those with the maximum score 1.0. The results are presented in Table 6. They are very similar to the results derived from using known labels in the test data for validation (Table 4), slightly better in some case since Precision@20 is likely to leave out low-quality predictions. It is therefore possible to use an external data source to verify made predictions. Spectral partitioning with m=0 and CN matching setting provides the best results, while Louvain with resolution=0.1 and CC matching setting does worst. Notably, the random walk baseline does worse for predictions on the test data than for those we evaluate using external information.

Table 6: Verification of found drug-target interactions on IUPHAR using external resources

Approach	Measure	Optimal parameters	Predictions verified, %	
			in total	with score 1.0
Baseline	-	eta=0.2, beta=0.7	77.58	45.33
Spectral partitioning	JC_{CC}	m=400, default threshold	78.65	42.90
	JC_{NC}	m=0, w/o threshold	83.23	47.65
Louvain algorithm	JC_{CC}	resolution=0.1	70.19	53.08
	JC_{NC}	resolution=0.8	82.41	44.06

Table 7: Example of verified (top) and non-verified (bottom) predictions using the UMLS as an external resource (* – not normalized)

Property	Value	Confidence
IUPHAR drug id	l1008	
Drug provisional name	sarafotoxin S6c	
Drug semantic types	[Amino Acid, Peptide, or Protein]	1000*
IUPHAR target id	t2206	
Target provisional name	phosphatase and tensin homolog	
Target semantic types	[Amino Acid, Peptide, or Protein]	1000*
	[Enzyme]	1000*
	[Molecular Function]	1000*
	[Qualitative Concept]	1000*
	[Amino Acid, Peptide, or Protein]	694*
	[Biologically Active Substance]	694*
	[Gene or Genome]	861*
	[Gene or Genome]	694*
	[Gene or Genome]	861*
Semantic type relations	[Amino Acid, Peptide, or Protein] interacts_with [Amino Acid, Peptide, or Protein]	0.694
	[Amino Acid, Peptide, or Protein] interacts_with [Amino Acid, Peptide, or Protein]	1.0
	[Amino Acid, Peptide, or Protein] interacts_with [Enzyme]	1.0
	[Amino Acid, Peptide, or Protein] affects [Molecular Function]	1.0
	[Amino Acid, Peptide, or Protein] interacts_with [Biologically Active Substance]	0.694
Confidence score	0.8776 x 5	
IUPHAR drug id	l1008	
Drug provisional name	sarafotoxin S6c	
Drug semantic types	[Amino Acid, Peptide, or Protein]	1000*
IUPHAR target id	t1495	
Target provisional name	phosphoinositide-3-kinase regulatory subunit 1	
Target semantic types	[Gene or Genome]	1000*
Semantic type relations	-	
Confidence score	0.0 x 0	

Table 8: Relation between network sizes and running times for IUPHAR and benchmark data sets

Data set	Quotient IUPHAR/benchmark set for $\|V\|^2$	$\|E\|$	Running times
Enzyme	92.03	82.98	59.27
GPCR	1119.3	894.61	871,62
IC	660.39	605.22	492.60
NR	17685.67	14467.41	23185.33
Kinase	435.17	263.73	349.88

1.3 Scalability

In order to verify scalability of our approach we tested it on the bigger data set, IUPHAR, and compared to running times on the benchmark data. Running times for a single test fold, the majority of which is taken up by community matching, are shown in Table 1 in the main paper. We show results for complete test folds since optimizing the parameter values via internal cross-validation will require five times this running time before the derived model can be applied to unseen data.

W.r.t. the number of vertices squared, i.e. the dimensionality of the matrices, dividing the value for IUPHAR by those of the benchmark data results in lower values than dividing running times, as Tabel 8 shows. The exception to this is NR, which is so small that it can be treated in less than a quarter second. Running times also grow more slowly than edge count, with the exception of Kinase, which might have to do with the fact that that data set is rather unbalanced, with far fewer drugs than targets. These results imply that the method scales, at least when using spectral partitioning.

In addition, Table 4 shows running times for the different combinations of community detection algorithm and matching method. Spectral partitioning is somewhat faster than Louvain but for node-community matching, this difference shrinks. Notably, even though node-community matching in itself is more expensive, the nature of this matching technique needs fewer communities, making up in terms of the community detection step itself.

REFERENCES

[1] O. Bodenreider. 2004. The unified medical language system (UMLS): integrating biomedical terminology. *Nucleic acids research* 32, suppl_1 (2004), D267–D270.

[2] D. Demner-Fushman, W. J. Rogers, and A. R Aronson. 2017. MetaMap Lite: an evaluation of a new Java implementation of MetaMap. *J. of the American Med. Inf. Association* 24, 4 (2017), 841–844.

[3] M. Koptelov, A. Zimmermann, and B. Crémilleux. 2018. Link Prediction in Multi-layer Networks and Its Application to Drug Design. In *IDA*. Springer, 175–187.