Deep Learning Approach to Link Weight Prediction

October 17, 2016

1 Experiments

We evaluate Model R experimentally and the results show that Model R can achieve much lower prediction error than the baseline models [1].

1.1 Datasets

The experiments use 4 datasets summarized in Table 1.

Table 1: The datasets used in experiments.

Dataset	Node count	Node type	Link count	Link weight type
Airport[2]	500	busiest airports in US	5960	number of passengers traveling from one airport to the other
Collaboration[4]	226	nations on Earth	20616	number of academic papers written by authors from the two connected nations
Congress[5]	163	102nd US Congress committees	26569	interlock value of shared members from the two committees
Forum[3]	1899	users of a student social network at UC Irvine	20291	number of messages sent from one student to the other

1.2 Experiment process

We do the identical experiment for each dataset. All the weights are normalized to ranger [-1, 1] after applying a logarithm function. Each experiment consists of 25 independent trials. In each trial, we split the dataset randomly into 3 subsets:

- 70% into training set
- 10% into validation set
- 20% into testing set

We use MSE (mean squared error) as the prediction accuracy metric of a trial. For each experiment, we report the mean and standard deviation of the 25 MSE's. The pseudo code of the experiment process is as follows:

```
def main():
    for dataset in [Airport, Collaboration, Congress, Forum]:
        (MSE_mean, MSE_standard_deviation) = do_experiment(dataset)
def do_experiment(dataset):
    MSEs = list()
    for trial in range(25):
        testing_error = append(evaluate_model_on(dataset))
        MSEs.append(testing_error)
    return (MSEs.mean(), MSEs.standard_deviation())
def evaluate_model_on(dataset):
    (trainning_set, validation_set, testing_set) = split(dataset)
    while validation_error decreases:
        training_error = estimator.learn(training_set)
        validation_error = estimator.predict(validation_set)
    testing_error = estimator.predict(testing_set)
    return testing_error
```

1.3 Experiment results

In our experiments, Model R's prediction error is lower than every other model on every dataset, shown in Figure 1. Now we compare Model R with only best baseline model - pWSBM. We calculate the error reduction from this best baseline model to Model R as:

$$Reduction = \frac{BaselineError - ModelRError}{BaselineError}$$

The reduction in prediction error is significant: ranging from 25% on Collaboration dataset to 73% on Airport dataset, shown in Table 2.

Table 2: The MSE's of 6 models on 4 datasets: Model R has lower error than every other model on every dataset, reducing error by 25% to 73% from the best baseline model - pWSBM. The number in every parenthesis is the standard deviation of MSE in 25 trials in the last digit of MSE.

Dataset	pWSBM	bWSBM	SBM	DCWBM	DCBM	Model R	Reduction
Airport	0.0486(6)	0.0543(5)	0.0632(8)	0.0746(9)	0.0918(8)	0.013(1)	73%
Collaboration	0.0407(1)	0.0462(1)	0.0497(3)	0.0500(2)	0.0849(3)	0.030(1)	25%
Congress	0.0571(4)	0.0594(4)	0.0634(6)	0.0653(4)	0.1050(6)	0.036(3)	35%
Forum	0.0726(3)	0.0845(3)	0.0851(4)	0.0882(4)	0.0882(4)	0.037(1)	48%

1.4 Computing resources

We ran our experiments on a Lenovo ThinkCentre M83 machine with the following specifications:

- Python implementation: CPython 3.5
- Operating system: Ubuntu 16.10 64-bit
- Memory: 16 GB
- Processor: Intel Core i7-4770 CPU @ 3.40GHz × 8

The program - coded in Python - uses all 8 threads of the processor. Each experiment takes about one hour to finish, depending on the dataset and parameters in the learning algorithm.

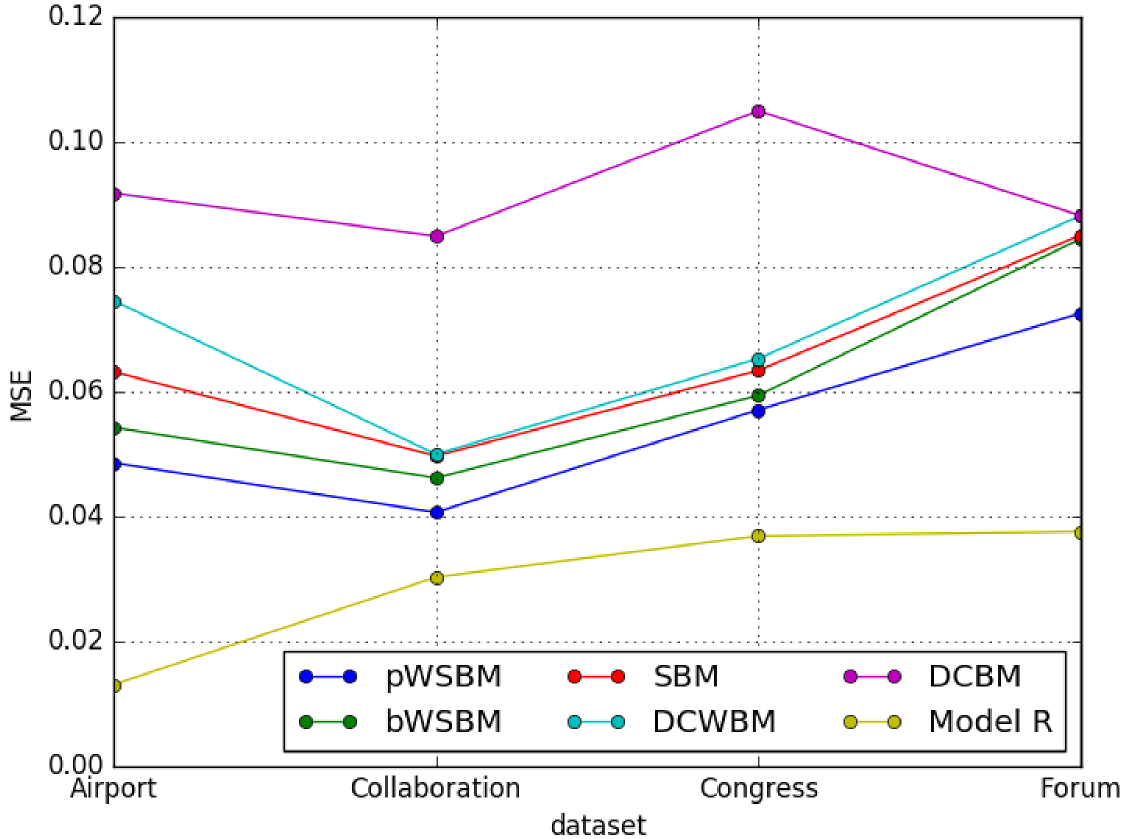

Figure 1: The MSE's of 6 models on 4 datasets: Model R has lower MSE than every other model on every dataset.

References

[1] Christopher Aicher, Abigail Z Jacobs, and Aaron Clauset. Learning latent block structure in weighted networks. *Journal of Complex Networks*, page cnu026, 2014.

[2] Vittoria Colizza, Romualdo Pastor-Satorras, and Alessandro Vespignani. Reaction–diffusion processes and metapopulation models in heterogeneous networks. *Nature Physics*, 3(4):276–282, 2007.

[3] Tore Opsahl and Pietro Panzarasa. Clustering in weighted networks. *Social networks*, 31(2):155–163, 2009.

[4] Raj Kumar Pan, Kimmo Kaski, and Santo Fortunato. World citation and collaboration networks: uncovering the role of geography in science. *arXiv preprint arXiv:1209.0781*, 2012.

[5] Mason A Porter, Peter J Mucha, Mark EJ Newman, and Casey M Warmbrand. A network analysis of committees in the us house of representatives. *Proceedings of the National Academy of Sciences of the United States of America*, 102(20):7057–7062, 2005.