Application of Machine Learning to Link Prediction

Kyle Julian and Wayne Lu
CS 229

Introduction

Link prediction in real-world networks can be found in recommendation engines such as for Amazon products and Facebook friends. Traditional link prediction approaches use topological and domain-specific heuristic values to rank pairs of unconnected nodes. We improve upon this method by training classification algorithms to generate rankings using only topological features.

Datasets

Dataset	Description	Nodes	Edges
wiki-Vote	Wikipedia RfA voting	3772	98978
ca-AstroPh	ArXiv AstroPh collab.	14175	189929
ca-CondMat	ArXiv CondMat collab.	14645	78785
ca-GrQc	ArXiv GrQc collab.	2155	9967
ca-HepPh	ArXiv HepPh collab.	7225	110243
ca-HepTh	ArXiv HepTh collab.	4306	17306
slashdot0811	Slashdot Zoo (Nov '08)	77360	905468
slashdot0902	Slashdot Zoo (Feb '09)	82168	948464

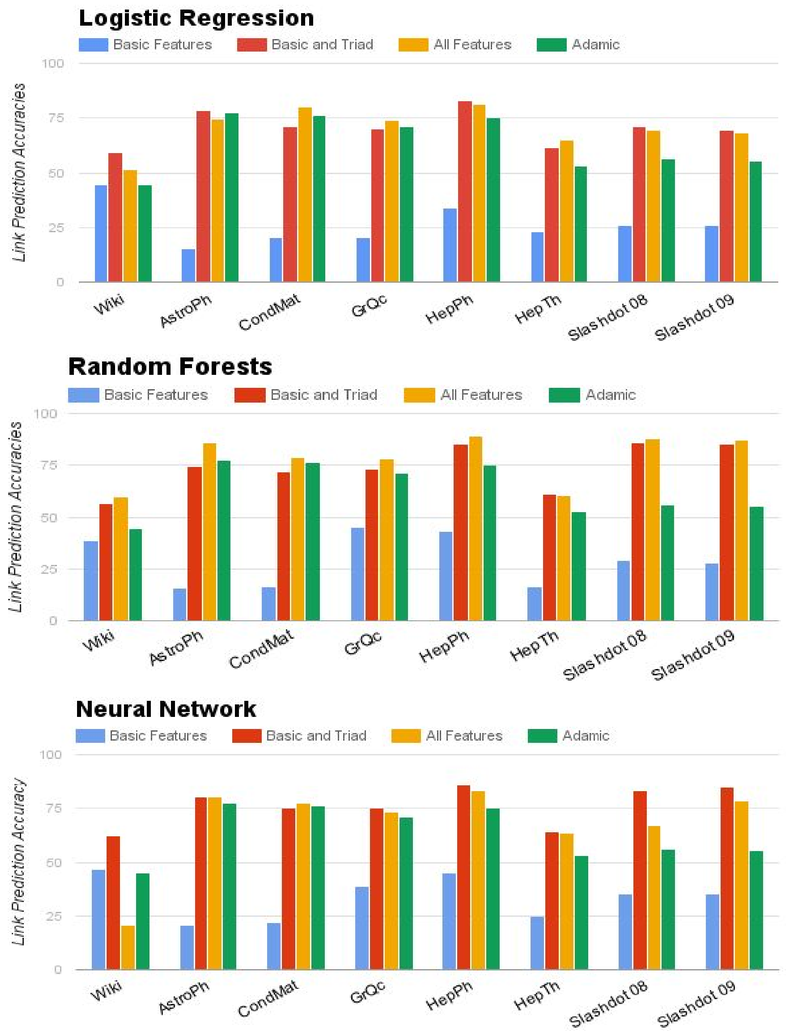
Features

For a pair of nodes (u, v), we extract the following features:

- 8 degree features: $d_{in}(u)$, $d_{out}(u)$, $d_{in}(u)/d_{out}(u)$, $d_{out}(u)/d_{in}(u)$, $d_{in}(v)$, $d_{out}(v)$, $d_{in}(v)/d_{out}(v)$, $d_{out}(v)/d_{in}(v)$
- 8 triad features: for a common neighbor w, we have 4 classes of triads based on edge direction. The features are the triad participation of (u, v) in the four triads and the ratios of participation to the number of common neighbors.

t_1	t_2	t_3	t_4
$u \rightarrow w \rightarrow v$	$u \rightarrow w \leftarrow v$	$u \leftarrow w \rightarrow v$	$u \leftarrow w \leftarrow v$

- 4 traditional link prediction heuristics: common neighbors, Adamic/Adar coefficient, Jaccard coefficient, and preferential attachment coefficient.



Methodology

For each dataset with graph G = (V, E):

1. Prune low-degree nodes
2. Randomly select 10% of E to form E' and remove them to create a graph G' = (V, E - E')
3. Randomly generate a set P of 9|E'| pairs of nodes (u, v) such that u and v share a neighbor, but (u, v) is not in E
4. Form a training set by sampling 70% of E' and P. Form a testing test by with the remaining edge pairs.
5. Extract features for the train/test sets and train the learning algorithms
6. Evaluate the accuracy of the learner on the test set

Accuracy Evaluation

Classification learners output a margin correlated to how sure the learner is about the classification. We sort the test set by this margin to find the k pairs of nodes which have the highest likelihood of being an edge, where k is the number of positive examples. The accuracy of the learner is then the proportion of those top k node pairs which are positive examples.

Cross Prediction

Each dataset was tested on a random forest trained on each of the datasets. Higher link prediction accuracies reveal similarity between datasets.