The Derivative Relational Attribute Graph Model for Link Prediction

W. A. Rivera, Ph.D.
and
Ryan Steed

Laboratory for Unconventional Conflict Analysis and Simulation*†
Social Science Research Institute
Duke University

August 16, 2017

Abstract

The derivative relational attribute graph (DRAG) model combines a standard random graph model with attribute affinity estimation to address the link prediction problem with a complex, comprehensive approach. Combining parameter estimates for relations at the attribute level and node interdependent, network-wide phenomena, the DRAG model encapsulates a breadth of network characteristics in a single link probability matrix that can be used to quickly and accurately perform such analyses as missing link identification, counterfactual analysis, and simulation. As a parameter estimator and a graph generator, we propose that the DRAG model is a powerful tool for insight into the rules that govern social interaction, represented by both the cumulative effects of interactions between core identities and the sweeping properties of networked communities.

Keywords: social networking, random graph model, network estimation, attribute affinity

*This material is based upon work funded by, or in part by, the Office of the Secretary of Defense with additional support from the U.S. Army Research Laboratory, the U.S. Army Research Office under grant number W911NF-15-1-0291.

†The views presented here belong solely to the author and are not necessarily the position of the Department of Defense or the U.S. Government.

1 Introduction

Complex *social networks* are extraordinarily valuable mechanisms for modeling social interactions in economic, social, and strategic spaces. Networks are exceptionally capable of representing a relationship between two entities, whether strategic social actors or abstract latent forces, as a link between nodes. Links can model information flow, sentiment, influence, or interpersonal relationships (Liben-Nowell and Kleinberg, 2007); links describe the essential elements governing social interactions in visual and mathematical terms. Further, the nodes in a social network may possess certain distinguishing attributes that comprise their identity and provide further variables for behavioral analysis. The study of complex social networks exposes the essential elements of social organization and uncovers emergent properties of unique combinations of people, places, resources, and ideas.

1.1 Background

Given the great advantages networks hold in the realm of social modeling, there have been many attempts to pursue robust link prediction methods that will extend network insights to counterfactual analysis, simulation, and the study of network evolution. Liben-Nowell and Kleinberg (2007) summarize the *link prediction problem*: "to what extent can the evolution of a social network be modeled using features intrinsic to the network itself?" Specifically, given an observed network in a fixed initial state, how does one predict the addition and removal of links, the mechanism for structural changes in a network, from the observed, initial network state to a final network state that represents the network's response to some external influence?

Many have answered this question with models that attempt to reproduce the broad structural features of real networks (Jin et al., 2001; Lusher et al., 2012; Fire et al., 2013). Others frame the link prediction problem as a method of statistical inference where a model is successful if it can accurately infer network evolution on a link-by-link basis (Newman, 2001; Dong et al., 2012). Computationally, researchers have identified key network statistics and measures that are useful for link prediction, but the prediction accuracy of simple predictive weight scores remains low even for a wide range of predictors (Fire et al., 2013; Liben-Nowell and Kleinberg, 2007). Some have turned to supervised machine learning

models based on key network features or statistics, using trained classification algorithms for link prediction in areas of research such as the security domain (Hasan et al., 2006; Budur et al., 2015; Krebs, 2002).

The field of graph generation can also be used as a method of link prediction; using certain network statistics, exponential random graph models (ERGMs) estimate statistic coefficients and create new network instances with statistical samples using those coefficients. The statistical validity of ERGM projections depends on the maximum likelihood estimation used; Shalizi and Rinaldo (2013) provide conditions for the stochastic projection of network parameters using an observed subset, which is usually a single graph or subgraph. Even with statistical checks, ERGMs using Markov Chain Monte Carlo algorithms, a common method of parameter estimation, often suffer the effects of *degeneracy* inference (Handcock, 2003). Chandrasekhar et al. (2014) attempt to address the degeneracy of inference problem with modified statistical treatments. Other advanced generative models have been developed to run realistic simulations or to generate extrapolations for cases in which there is a dearth of data, such as the Kronecker product model (Leskovec et al., 2005). However, few generative models successfully incorporate the impact of node attributes on links for a variety of network types and scales. Kim and Leskovec (2011) use the concept of attribute affinities to describe the cumulative effect of attribute relations on node relations, developing an efficient Multiplicative Attribute Graph (MAG) Fit algorithm that outperforms the Kronecker product model in coefficient estimation for several network properties. For networks with rich attribute data, the identity of node, as described by its attributes, is an extremely useful basis for link prediction.

1.2 Problem Statement

We seek to develop a computationally efficient link prediction model for simulating the reaction of an observed network to an external influence by exploiting intrinsic network properties. Specifically, we seek to combine the cumulative effect of many attribute relations on relationships with broad, structural network properties to produce a comprehensive, generalized application of link prediction techniques for social analysis.

2 Solution

2.1 Approach

Our approach, the Derivative Relational Attribute Graph (DRAG) model, combines a version of standard network parameter estimation using an ERGM for network-statistic estimation with the MagFit algorithm for attribute affinity estimation (Kim and Leskovec, 2011). By converting parameter estimates for both relations at the attribute level and node interdependent, network-wide phenomena, the DRAG model captures a breadth of network characteristics in a single link probability matrix. Examined as a measure of link prevalence or combined with a generative model to produce likely network instances in response to an external influence, the DRAG model is a powerful tool for link prediction and social insight. The following sections summarize the MagFit algorithm, our proper ERGM model, and the combination of the MagFit algorithm and a proper ERGM algorithm into a single posterior probability adjacency matrix used for graph generation and link prediction.

2.2 Multiplicative Attribute Graph Model

What is the probability that an edge between two specific nodes will form, given a defined set of affinities between attributes? The multiplicative attribute graph (MAG) model defined by Kim and Leskovec (2011) is a parameter estimation technique for node-node attribute relations and is used to capture network relational properties such as homophily, affiliation, and aversion in an affinity matrix for categorical attribute combinations in a network. Using an observed, directed network of nodes with an attribute vector, the MagFit algorithm produces an estimated affinity matrix that can be used to calculate the probability of any possible link.

For N nodes in directed graph Y, each node i has A categorical attributes, $F_{i1} \cdots F_{iA}$. Each attribute a ($a = 1, \cdots A$) is associated with affinity matrix Θ_a. For n possible attribute values, each entry $\Theta_a[k, k'] \in (0, n)$ of the affinity matrix indicates the potential for a pair of nodes to form a link, given the a-th attribute value k of the first node and value k' of the second node. For a given pair of nodes, the corresponding attribute values denote a set of affinities from the affinity matrices for each attribute. The link probability is the product

of the selected values. Each edge (i, j) is included in network Y with the probability $p_{i,j}$ (Kim and Leskovec, 2011):

$$p_{i,j} := Pr(Y_{i,j}) = \prod_{a=1}^{A} \Theta_a[F_{ia}, F_{ja}]$$

Thus, the MAG model can produce a matrix P of edge probabilities $p_{i,j}$ based on the attribute affinity matrix. To reverse this process and estimate the affinity matrix Θ_a, the probability matrix P is used to find the maximum likelihood estimate according to the following likelihood function, where Y is expressed as an adjacency matrix (Kim and Leskovec, 2011):

$$P(Y|F, \Theta) = \prod_{Y_{i,j}=1} p_{i,j} \prod_{Y_{i,j}=0} (1 - p_{i,j})$$

However, due to the high variance and combinatorial problems associated with maximizing this function, a simple generative model is applied to the node attributes such that the a-th attribute takes value 1 according to a Bernoulli probability distribution (Kim and Leskovec, 2011):

$$F_{ia} \sim Bernoulli(\mu_a)$$

Since approximation sets each edge independent given attributes F and sets each attribute independent given parameters μ_a, we can obtain the final likelihood function:

$$P(Y|F, \Theta)P(F|\mu) = \prod_{Y_{i,j}=1} p_{i,j} \prod_{Y_{i,j}=0} (1 - p_{i,j}) \prod_{F_{ia}=0} \mu_a \prod_{F_{ia}=1} (1 - \mu_a)$$

After optimizing the equation and adding a penalty term to maximize entropy, the MagFit algorithm converges to estimates for parameters μ and Θ (Kim and Leskovec, 2011).

2.3 Exponential Random Graph Model

The proper exponential random graph model provides us with the means to estimate the coefficients associated with any network statistic. If we change the structure of the network,

how do the network statistics change in response? How does the posterior probability of observing a particular network change? Once we know the importance of each statistic to our network's character, we can pick the edges that fit our network's structure best. The only knowledge we need beforehand is the adjacency matrix (the current structure) of our network and what measures might be salient to our network's structure, and how much we want it to retain that structure.

In this model, the ERGM is used as a generative model that produces likely adjacency matrices based on attribute affinities and other network characteristics, expressed as network statstics. The ERGM balances the probability matrix produced by the MAG model affinity matrix estimate. It maintains network statistics by ensuring that new links best preserve the character of the network, where the character is composed of the statistics we choose, minimizing network degeneracy. Further, an ERGM preserves emergent properties of a network that go beyond node-node interactions and independent relationships, capturing network transitivity and the interdependencies of network formation (Chandrasekhar et al., 2014).

If Y is a network adjacency matrix, θ is a vector of coefficients, and the function $\Psi(Y)$ produces a vector of network statistics on that adjacency matrix, the probability of a particular adjacency matrix Y given a vector of coefficients θ can be estimated by the following:

$$Pr(Y|\theta) \propto e^{\theta' \Psi(Y)}$$

$\Psi(Y)$ is represented by a vector of any combination of network statistics Φ_k. We propose the following as useful statistical measures for link prediction:

- Transitive friends count (Fire et al., 2013)
- Preferential attachment count (Fire et al., 2013)
- Katz measure: a more robust variation of the shortest path measure that is used in many supervised learning models (Hasan et al., 2006)
- Mutual tie count: number of reciprocative ties

- In-density (1-instar count): number of edges in the network
- Out-density (1-outstar count): number of edges in the network
- 2-instar count: number of nodes that are the target of two links
- 3-instar count: number of nodes that are the target of three links
- 2-outstar count: number of nodes that are the source of two links

Specifically, each covariate is the change in one of the statistics above for two cases: the case in which the edge in question is present in the graph, and the case in which it is not. Let $Y^+_{i,j}$ be network structure Y when $y_{i,j} = 1$ and let $Y^-_{i,j}$ be network structure Y when $y_{i,j} = 0$. For network statistic $\Phi_k(Y)$ the change statistic for the edge (i, j) is defined as follows:

$$\phi_k(Y, i, j) = \Phi_k(Y^+_{i,j}) - \Phi_k(Y^-_{i,j})$$

Thus, the function $\Psi(Y)$ produces a vector of change statistics ϕ_k for each network statistic Φ_k. The vector of coefficients θ' is simply a vector of random variables θ'_k represented by a normal distribution centered on zero:

$$\theta'_k \sim N(\mu, \sigma)$$

To find the actual probability of the network Y, the calculated conditional probability above must be normalized against the probabilities of all possible network configurations:

$$Pr(Y) = \frac{Pr(Y|\theta)}{\sum_{y \in Y} Pr(y|\theta)}$$

For even the simplest network, there are a very high number of possible configurations that must be calculated to achieve a perfectly normalized statistic. To address the computational cost of assessing a high number of network permutations, Monte Carlo metropolis chain (MCMC) algorithm using a Gibbs sampling method takes the matrix of probabilities $Pr(Y|\theta)$ for each individual edge resulting from the product of the vector of coefficients θ' and the vector of network statistics $\Psi(Y)$ and uses it to estimate the normalized coefficients for network Y given all possible networks y (Brooks, 2010). Specifically, the MCMC

explores the most probable areas of the parameter space (where the parameters are the covariates defined above), sampling the areas of the parameter space where conditional probabilities are high and ignoring the areas where conditional probabilities are low.

Since a Bernoulli distribution is a binomial distribution with only a single trial, a matrix of these random variable distributions will contain only 0s and 1s, successes and failures, edge presences and edge absences. Thus, the matrix of random variables O can be realized as the adjacency matrix for an observable network. The MCMC model is instantiated with the matrix of random variables O as the observed graph to fit (our priors) and the list of coefficients as the parameters to estimate. For a set number of iterations, the MCMC returns the optimized (maximum likelihood) coefficient values and the probability of observing the adjacency matrix O given the set of fitted parameters θ. To produce a random instance of the network, a random realization from the posterior distribution can be drawn after fixing the optimized coefficients.

2.4 Derivative Relational Attribute Model (DRAG)

The DRAG model combines the probability matrix output of these two models into single generative model for link prediction. This model will take an observed network, containing nodes with attributes, and produce a single, combined link probability adjacency matrix based on parameter estimation by both the ERGM and the MAP model. By combining both the models above, we produce a new network that reflects both the original system characteristics and the interaction between roles and attributes we seek to model.

From the proper ERGM, we obtain a matrix of normalized edge probabilities using the equation below, where the vector of coefficients θ has been estimated by a MCMC algorithm:

$$Pr(Y|\theta) \propto e^{\theta' \Psi(Y)}$$

From the MAG model, we can obtain a matrix of non-normalized edge probabilities based on the following equation, which takes an attribute linking affinity matrix as its input:

$$p_{i,j} := Pr(y_{i,j}|\Theta_a) = \prod_{a=1}^{A} \Theta_a[F_{ia}, F_{ja}]$$

We must now calculate a probability for each network sampled in the MCMC model given the set of fitted coefficients determined by the MCMC model and the attribute linking affinity matrices. Two weighting coefficients will be applied to prevent the assertion of one model over the other; these coefficients can be determined with empirical testing or left up to an analyst's discretion. The best method for determining weight coefficients is network-specific algorithm, dependent on the data available, for the express purpose of maximizing accuracy: for networks with high attribute data saturation, c_a is relatively higher; for networks with strong structural statistics or very complex structures, c_p is relatively higher. Such a method provides a flexible balancing mechanism for the outputs of each model component.

Before balancing the output matrices, we must convert the output of the ERGM to a probability matrix like the one produced by the MAP model. Given the MCMC-maximized vector of coefficients θ, the probability of any edge can be estimated as follows, given that the ERGM output can be re-expressed as the log-odds of observing a particular edge in network y:

$$p_{i,j} := Pr(y_{ij}|y_{-i,j}, \theta) = logistic \sum_{h=1}^{k} \theta_h \Psi_h^{i,j}(Y)$$

Now that both model outputs are in the same form, they can be combined with the weighting coefficients c_p and c_a, where p denotes the proper ERGM and a represents the MagFit algorithm output:

$$p_{i,j\,\text{DRAG}} = \frac{c_p p_{i,j_p} + c_a p_{i,j_a}}{c_p + c_a}$$

2.5 Network Generation

To convert the probability matrix into a realized network, another Bernoulli approximation is used to generate a value of 0 or 1 for every potential edge in the network such that an adjacency matrix X represents a likely network realization.

$$X_{i,j} \sim Bernoulli(p_{i,j\,\mathrm{DRAG}})$$

Thus, once the probabilities p_{DRAG} for each edge are converted to outcomes X with some element of randomness, the model outputs an adjacency matrix for a new network instance that reflects both the original system characteristics and the interaction between roles and attributes.

3 Conclusion

3.1 Applications

The DRAG model is most usefully applied to simulating network response to external influence, such as link addition, link removal or node attribute changes. Common heuristics for predicting node interactions in a network, such as transitivity and reciprocity, are included in the estimated statistics for the ERGM portion of the model, while heuristics like homophily and repulsion are inherent to attribute affinities. The probability matrix output can be used to examine the relative likelihood of links for a new node, links generated in response to a network restructure, or altered potential link likelihoods in response to an attribute change; this is useful for identifying "missing" links in a network that are not empirically observed but are highly likely or for finding anomalies between observed actor behavior and likely actor behavior. The probability matrix can also be used to sample new network realizations for simulations or analyses. If the DRAG model is combined with a set of rules that change attributes or attribute weights in response to network conditions, a self-sustaining simulation can be run without analyst interference; as new networks are generated and attributes are altered, probability outputs change and new networks are generated. The effects of a single action can cascade over countless iterations until the network stabilizes at some equilibrium that can be used to draw strategic conclusions about counterfactual scenarios or future projections.

Another specific advantage of producing a probability matrix is that it can be modified after the computation section of the model. If there are particular role behaviors or interactions that are not captured by affinities, there are conditions in which edges must or must

not exist, or an analyst has specific *a priori* knowledge about the rules governing social interaction in a particular network, the probability matrix can be adjusted post-estimation with programmatic routines appended to the main algorithm. Such a modification allows an analyst or researcher to assert specific knowledge of social rules over a system without sacrificing data-driven conclusions.

The DRAG model can be applied to any network analysis application, especially in situations where attribute data is plentiful. In contrast to highly advanced and complex methods of graph generation, the DRAG model utilizes a fairly simple structure with a relatively low computational expense. Further, the combination of traditional, network-wide statistical parameters with specific attribute affinities allows the DRAG model to make more realistic predictions for a wide variety of observable networks with varying sizes and data saturation.

A fast, scalable, accurate link predictor model is a valuable asset for applications in a variety of fields: social media friendship networks, strategic analysis in the security domain, bioinformatics insights in areas such as protein interaction, scientific collaboration networks, and even artificial intelligence.

3.2 Future Research

To improve the accuracy of attribute relational estimation, the DRAG model could also incorporate an infinite multiple membership relational model (IMRM) like the ones used for community identification (Mørup et al., 2011). By using a non-parametric latent feature model, the DRAG model could incorporate positive and negative attribute weights if attributes are treated as membership vectors. However, current multiple membership relational models have high CPU runtimes and would require optimization for an applicable analysis tool.

To utilize the IMRM in the DRAG model, a Hamiltonian Markov chain Monte Carlo using Gibbs sampling and split-merge moves must be used instead of the current MCMC (Mørup et al., 2011).

The DRAG model must also be optimized to handle network structures with missing data for the cases in which attribute affinities are inaccurate due to incomplete attribute

data. In this case, the weighting mechanism must be adjusted to favor the ERGM probability matrix over that of the MagFit algorithm.

4 Algorithms

4.1 MAG

The DRAG model utilizes the "fast" MagFit algorithm proposed by Kim and Leskovec (2011).

4.2 ERGM

The DRAG model also utilizes the open-source PyMC Python package for Bayesian statistical analysis (Fonnesbeck et al., 2015). Specifically, we apply the Normal and Bernoulli distribution functions for coefficient prior probability distributions and graph generation, respectively. We also apply the built-in stochastic deterministic decorator for likelihood function probability calculations and the PyMC Markov Chain Monte Carlo model for model construction and random walk sampling. The ERGM model takes a NetworkX Python object as its input, and returns a posterior probability adjacency matrix (Hagberg et al., 2010). The ERGM section takes the standard format of a Markov Chain Monte Carlo algorithm, first generating prior probabilities on a Normal distribution, then calculating likelihood for the initial state, then triggering the Monte Carlo sampling algorithm for approximately 50,000 iterations.

References

Brooks, S. P. (2010), "Markov Chain Monte Carlo Method and its Application," *Journal of the Royal Statistical Society*, 47, 69–100.

Budur, E., Lee, S., and Kong, V. S. (2015), "Structural Analysis of Criminal Network and Predicting Hidden Links using Machine Learning," *CoRR*, abs/1507.0.

Chandrasekhar, A. G., Jackson, M. O., Andrews, I., Blume, L., Carroll, G., Chernozhukov, V., Duflo, E., Golub, B., Graham, B., Honore, B., Lewis, R., Mele, A., and Morales, E. (2014), "Tractable and Consistent Random Graph Models," *National Bureau of Economic Research*.

Dong, Y., Tang, J., Wu, S., Tian, J., Chawla, N., Rao, J., and Cao, H. (2012), *Link Prediction and Recommendation across Heterogeneous Social Networks*.

Fire, M., Tenenboim-chekina, L., Puzis, R., Lesser, O., Rokach, L., and Elovici, Y. (2013), "Computationally Efficient Link Prediction in a Variety of Social Networks," *ACM Transactions on Intelligent Systems and Technology*, 5.

Fonnesbeck, C., Patil, A., Huard, D., and Salvatier, J. (2015), "PyMC 2.3.6," .

Hagberg, A., Schult, D., and Swart, P. (2010), "NetworkX," .

Handcock, M. S. (2003), "Statistical Models for Social Networks: Inference and Degeneracy," in *Dynamic Social Network Modeling and Analysis: Workshop Summary and Papers*.

Hasan, M. A., Chaoji, V., Salem, S., and Zaki, M. (2006), "Link Prediction using Supervised Learning," in *In Proc. of SDM 06 workshop on Link Analysis, Counterterrorism and Security*.

Jin, E. M., Girvan, M., and Newman, M. E. J. (2001), "The structure of growing social networks," *Physical Review E*, 64.

Kim, M. and Leskovec, J. (2011), "Multiplicative Attribute Graph Model," *Internet Mathematics*, 8, 113–160.

Krebs, V. E. (2002), "Mapping Networks of of Terrorist Terrorist Cells," *Connections*, 24, 43–52.

Leskovec, J., Chakrabarti, D., Kleinberg, J., and Faloutsos, C. (2005), "Realistic, Mathematically Tractable Graph Generation and Evolution, Using Kronecker Multiplication," in *9th European Conference on Principles and Practice of Knowledge Discovery in*

Databases, eds. Jorge, A. M., Torgo, L., Brazdil, P., Camacho, R., and Gama, J., Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 133–145.

Liben-Nowell, D. and Kleinberg, J. (2007), "The Link Prediction Problem for Social Networks ," *Journal of the American Society for Information Science and Technology*, 58, 1019–1031.

Lusher, D., Koskinen, J., and Robins, G. (eds.) (2012), *Exponential Random Graph Models for Social Networks: Theory, Methods, and Applications*, Cambridge University Press.

Mørup, M., Schmidt, M. N., and Hansen, L. K. (2011), "Infinite Multiple Membership Relational Modeling for Complex Networks," in *IEEE International Workshop on Machine Learning for Signal Processing*.

Newman, M. E. J. (2001), "Clustering and preferential attachment in growing networks," *Physical Review E - Statistical, Nonlinear, and Soft Matter Physics*, 64, 251021–251024.

Shalizi, C. R. and Rinaldo, A. (2013), "Consistency under sampling of exponential random graph models," *The Annals of Statistics*, 41, 508–535.