Playing Multiple Roles: Discovering Overlapping Roles in Social Networks

Alicia P. Wolfe PIPPIN@CS.UMASS.EDU
David Jensen JENSEN@CS.UMASS.EDU
140 Governor's Drive, University of Massachusetts, Amherst, MA 01003-4610 USA

Abstract

Many social networks can be characterized by the roles of the participants in the network. Participants with similar roles exhibit similar patterns of link structure. For example, a dentist has links to patients and other dentists, a PTA member has links to parents and students at a local school. We extend existing methods of finding roles to include multiple role labels (e.g., dentists who are also PTA members), and apply Gibbs sampling methods to find the maximum likelihood role labels and link probabilities on an unlabeled graph. We use synthetic data to evaluate the accuracy of this method compared to methods which assume a single role labeling.

1. Introduction

1.1. Why find roles?

The idea of modeling graph structure in social networks by finding "roles" of nodes in the graph was first introduced by Lorrain and White [13]. The basic idea is that two entities have the same role if they both are related to other entities in the same pattern (see figure 1). For example, a dentist usually has links to patients and hygenists, where a PTA (Parent-Teacher Association) member generally has links to parents and students at a local school. Ideally, good role labels on nodes in the graph would provide sufficient information to predict the link structure of the graph.

Many algorithms look for specific type of structure in the graph – highly connected subcomponents, or clusters. While clusters are interesting, we will focus instead on the *role* of a node. In some cases, the label being found does not cluster, for example, dentists are in general more connected to their patients and hygenists than other dentists. Note, however, that a cluster can be represented as a role with high probability of linking to itself and low probability of linking to other roles.

There are many precise mathematical definitions of what it means to have the same role, stochastic equivalence is the version used here:

> Let X be a random adjacency array. We say two nodes i and i' are stochastically equivalent if and only if the probability of any event about X is unchanged by interchanging nodes i and i'. [9]

In the simplest case "events" consist only of the links between nodes. In this case, two nodes are equivalent if they have the same probability of linking to nodes of each role label.

While attribute information can be added to this model, in its simplest form it only attempts to capture the link structure of the graph. We will use this fact to explore the effects of changes in the link generation model on role label prediction, in order to carefully isolate and analyse the effects of modifying the model to include multiple role labels.

Role induction allows the creation of a small abstract model of our graph. The model is a homomorphic image of the original graph, which contains summary information about the role labels and their relationships [17], as shown in figure 2. This model can be used to classify nodes into role types, predict links between nodes based on role label, summarize the properties of relationships among nodes, make predictions about similar graphs, and compare graphs to each other.

1.2. Multiple Role Labels

Existing work in social networks on finding roles assumes that an entity plays only one role in the graph. However, many situation require multiple roles to adequately model link structure. For example, a dentist may also be a PTA member, with links generated by

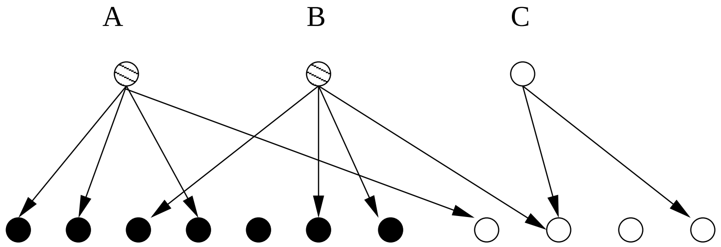

Figure 1. Nodes A and B have similar links to other nodes, and therefore the same role, whereas C shows a different link pattern. Only links connected to A, B and C are shown.

both roles. Methods which attempt to find only one set of role assignments in such a graph may be misled by the overlapping labels and end up finding neither correctly. If the number of labels used in the algorithm is appropriate for just one of the sets of role labels, the "noise" from the alternate role label can obscure both. For example, one might approximate the number of careers at 8 and attempt to find them, however, if there are 4 volunteer position roles which are also affecting the generation of links, the effective number of roles in the network is 32.

One adaptation of existing techniques to this problem would be to increase the number of role labels until each combination of roles has its own label, using the full 32 labels in our example. However, while this can find accurate results, it loses important information – we would really like to find a group containing all dentists, not all dentist-PTA members. This requires multiple labels on each node.

Using multiple labels provides a second benefit, as well, due to the overlapping labels on each node. The data provided by the graph is used more efficiently by this model, which means it can be learned from smaller graphs.

2. Modeling Multiple Roles

We assume that roles are separated into catagories, and that each entity has one label from each catagory. In our earlier example, one catagory would be "career" (dentist) another might be "volunteer position" (PTA member). It is always possible to include a "null" placeholder label in a catagory, if, for example, not all entities have volunteer roles.

We also assume that links are generated by each role catagory independently – there is a set of links generated by the fact that an entity is a dentist, another by the fact that they are a PTA member. The total set of links for an entity is the union of all such sets of generated links. The total probability of a link between two nodes is therefore the "or" of the probabilities that the role labels in each type generated a link.

It is therefore straightforward to calculate the total probability of a link between two nodes u and v with composite role labels I and J. If there are T role types, composite label $I = \{i_0, i_1, ...i_T\}$, and $J = \{j_0...j_T\}$. Label the probability of a link being generated between u and v by the t^{th} role label elements i_t and j_t, p_{i_t,j_t}. These p_{i_t,j_t} values will be the parameters of our model. The total probability of a link from node u to v is the "or" of the probabilities that each role type t generated a link:

$$p_{IJ} = 1 - \prod_{t \in T} 1 - p_{i_t,j_t}. \tag{1}$$

The parameters p_{IJ}, along with priors on the role labels, can be used to find the total probability of a particular labeling, parameters, and graph structure by combining the likelihood of each edge (present or absent) in the graph.

A graph G consists of V, the set of nodes, and E, the edge matrix with non-zero entries $e_{uv} = 1$ if and only if there is a link from node u to node v. If $\mathcal{R}$ is the set of role labels, where $R_u \in \mathcal{R}$ is the label for node u, then the joint probability of the entire graph $G = (V, E)$ with labeling $\mathcal{R}$ and parameters $\Theta = \{P(R_u)\} \cup \{p_{i_t,j_t}\}$ is:

$$p(G, \mathcal{R}, \Theta) = \prod_{u \in V} \left[P(R_u) \cdot \prod_{v \in V} pbinom(e_{uv}, p_{R_u,R_v}) \right]$$

where pbinom(sample, probability) is the probability of a particular sample of 0's and 1's being drawn from a binomial distribution with the specified probability.

This model can be used find the maximum likelihood role labelling and parameters given a graph, using Gibbs sampling.

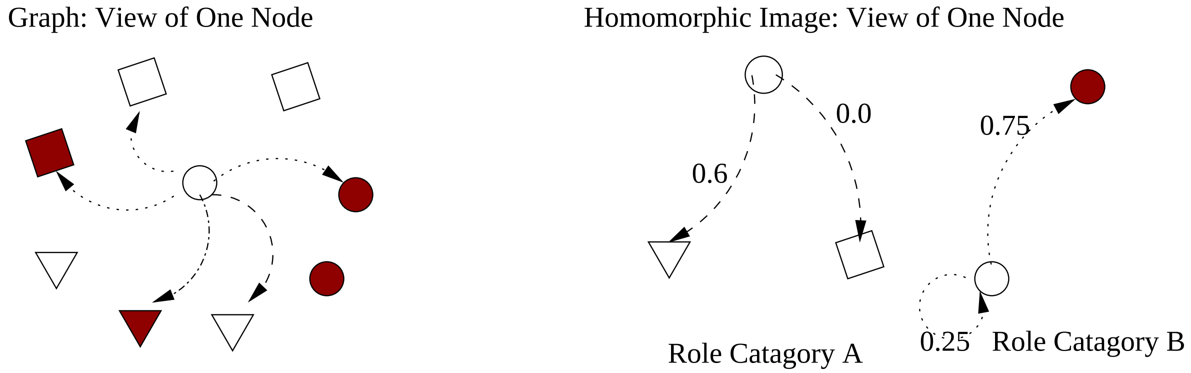

Figure 2. A multiple roles model of a graph. The image of the graph includes two types of role, one with 3 role labels, and one with two. Links can be generated by either role type.

3. Finding Multiple Roles

Snijders and Nowicki [16, 14] tackle the problem of finding both role labels and parameters for a graph with a single set of labels. They use EM, and, when that becomes intractible for larger graphs, Gibbs sampling. The unknown variables are the role labels of the nodes, and the parameters of the model are the link probabilities between role labels and priors for role labels. We extend their method by including multiple role labels using the model (and modeling assumptions) from the previous section.

Given an unlabeled graph G, the goal is to find the combination of role labels and parameters which is most likely. Gibbs sampling is one effective method of doing so, though as with many algorithms it may end up in a local extrema.

The entire algorithm is separated into two phases: burn in and sampling. Each phase consists of many incremental iterations of model estimation, each of which generates a sample labeling of the graph. The iteration steps in each of the two phases are identical, however, the samples generated by the burn in phase are assumed to be biased by the initial settings of the parameters, and are discarded. The sample labelings generated in the second phase are used to generate probabilities over labels for each node. The number of burn in and sampling steps necessary depends on the properties of the specific graph and model.

```
procedure $Gibbs - Sampling - Algorithm$

  Initialize parameters randomly
  Initialize labels randomly
  for numBurnInIters do
    RelabelGraph
  end for
  for numSampleIters do
    RelabelGraph
    for all $v \in V$ do
      $count_{v,R_v} = count_{v,R_v} + 1$
    end for
  end for
  for all $v \in V$ do
    for all $I \in \mathcal{R}$ do
      $P(R_v = I) = count_{v,I}/numSampleIters$
    end for
  end for
```

```
procedure $RelabelGraph$
  Part 1: Sample Labels
  for all $v \in V$ do
    $R_v$ = sample from $P(R_v = I|\{R_u, u \neq v\}, \Theta)$
  end for
  Part 2: Maximum Likelihood Parameters
  for all $I \in \mathcal{R}$ do
    $P(I) = argmax_{P(I)} P(G, V, \mathcal{R}, \Theta)$
    for all $J \in \mathcal{R}$ do
      for $t$ from 1 to $T$ do
        $p_{i_t,j_t} = argmax_{p_{i_t,j_t}} P(G, V, \mathcal{R}, \Theta)$
      end for
    end for
  end for
```

The Gibbs sampling algorithm is a general framework which can be applied with a variety of models. The algorithm can be adapted to a specific model (in our case the model from section 2) by inserting the appropriate formulae for the conditional distributions for the missing variables ($\mathcal{R}$), as well as the maximum likelihood values of the parameters (Θ).

3.1. Labeling Nodes

In the first part of each iteration, the algorithm generates a new value for each missing variable (role labels in this case) by sampling from the conditional distribution of the variable, given the rest of the graph, with its current labels. Since each node's role label is independent of the labels on nodes more than one link away, the conditional for the composite role label a single node v can be written as:

$$\begin{aligned}&P(R_v = I|\{R_u, u \neq v\}, \Theta) \propto \\ &\quad P(I) \cdot \textstyle\prod_{u \in V} pbinom(e_{uv}, p_{R_u,I}) \cdot pbinom(e_{vu}, p_{I,R_u})\end{aligned}$$

The algorithm samples from this conditional distribution to generate a new label for each node in turn until the entire graph has been re-labeled.

3.2. Maximum Likelihood Parameters

Part 2 of each Gibbs sampling iteration calculates the maximum likelihood value of our model parameters, given a labeled graph. The parameters of our model consist of the link probabilities for each role pairing, and the priors for role labels. The priors for role labels can be estimated directly from counts over the data:

$$\begin{aligned}c_{R_k} &= \text{number of nodes labeled} R_k \\ n &= \text{number of nodes} \\ \hat{P}(R_k) &= c_{R_k}/n\end{aligned}$$

The remaining parameters to estimate are the link generation probabilities. Conveniently, these are factored according to the role elements. This will enable us to make more efficient use of the data in our social network, since each probability estimation for a role element includes information from a larger subset of the nodes in the network than if flat role labels were used. However, it also complicates our calculation of expected link probability values.

In the case where there is a single set of role labels, the new value of each link probability is easily calculated from simple counts. If c_{IJ} is the count of links between nodes of role I and nodes of role J, and n_{IJ} is the number of pairs of nodes labeled (I, J), the estimated link probability is simply $\hat{p}_{IJ} = c_{IJ}/n_{IJ}$.

However, finding the parameters when there are multiple role labels is slightly more complex. Consider two nodes, u and v. If there are T role labels on each node, each of the links between the nodes could have been generated by any one (or more) of the T pairs of role label elements on the nodes, as shown in figure 2. If a dentist/PTA member has a link to a hygenist/student, we must decide how much of the responsibility for the link to assign to the dentist-hygenist role pair, and how much to the PTA-student role pair.

This value is not immediately available, though it could be calculated from the actual parameters for the model. Instead, we create T "missing" variables on each link, $\{m_0, ...m_T\}$. Each m_t contains an estimate of the probability that a particular pair of role labels (in catagory t) generated the link.

Each m_t can be estimated from the parameter estimates from the last iteration. If u and v are the nodes at the ends of the link, and I and J are their role labels, then:

$$m_t = \hat{p}_{i_t, j_t} / \hat{p}_{IJ}$$

When the new value for $\hat{p}_{i_t, j_t}$ is calculated, all links between nodes that contain the labels i_t and j_t must be considered. If each link is weighted with the probability that it was generated by i_t and j_t, and the sum over all links taken, the result is a count of the number of links that should be attributed to the i_t and j_t pair. Dividing number of such pairs gives us the new link probability:

$$\begin{aligned}c_{i_t,j_t} &= \sum_{IJ|i_t \in I \wedge j_t \in J} m_t \cdot c_{IJ} \\ &= \sum_{IJ|i_t \in I \wedge j_t \in J} \hat{p}_{i_t,j_t}/\hat{p}_{IJ} \cdot c_{IJ} \\ \hat{p}'_{i_t,j_t} &= c_{i_t,j_t}/n_{i_t,j_t}. \qquad (2)\end{aligned}$$

$\hat{p}_{IJ}$ is simply a function of the $\hat{p}_{i_t,j_t}$ values, as we saw in equation 1.

An alternative solution to this problem would be to create unknown labels on each link indicating the (hypothetical) set of role catagories which generated it. The sampling phase of the Gibbs sampler would then generate these labels, and counts could be taken over the set of links which have been labeled as being generated by a particular role catagory.

3.3. Reintroducing Lost Labels

If at the end of an iteration a label no longer appears anywhere in the graph, it is assigned to a random node before the begining of the next iteration.

4. Results on synthetic data

4.1. Synthetic Experiments

For our synthetic data experiments, graphs were constructed based on our modeling assumptions from sec-

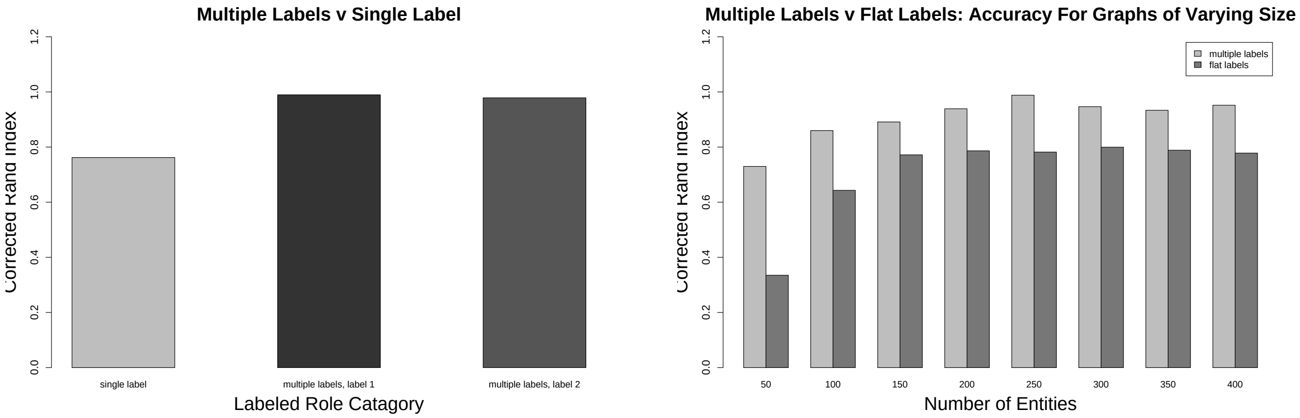

Figure 3. In the left chart, we can see the difference that ignoring one or more of the role catagories can make. For a graph with C role catagories, the multiple roles labeler was run with C catagories, and the results matched to the true labeling. It was then run with only 1, and again compared to the true labels. CRI values are the best match with any one of the C true role catagories, averaged over 20 runs. In the right chart, we see the advantage of using multiple roles over flattened roles when the data set is small. The multiple role labeler finds the true role structure, two roles with 3 values each, while the flat role labeler uses one role catagory with 9 values. CRI values are calculated with respect to the "flattened" correct role labelling, averaged over 20 runs.

tion 2. These graphs were used to evaluate the performance of the multiple roles Gibbs sampling algorithm and compare it to the same algorithm using a single role label. When the number of labels in each role catagory is the same, our experiments performed as expected, allowing the algorithm to find more accurate, expressive models and requiring less data to do so. However, there were also interesting problems when the number of values in different role catagories was not uniform, as we will see in section 4.5.

Each generated graph was constructed with a fixed number of role catagories, and a fixed number of role values in each catagory. Role priors and link generation probabilities were generated uniformly between 0.0 and 1.0. The graph labels and links were then assigned according to the model in section 2.

4.2. Evaluating Results on Labeled Data: the Corrected Rand Index

Since these experiments were run on generated data, the correct role labels, which generated the graph, were available. Therefore the roles found were evaluated with respect to this "correct" labeling using the Corrected Rand Index.

The Rand Index [15] evaluates the percentage of pairs of vertices which are correctly located either in the same set or different sets. [11] correct this measure for chance, with a resulting measure which has a mean of 0 for a randomly chosen partition, and a maximum value of 1.0 (complete agreement). If we define, for a correct labeling C and algorithmic labeling A:

$$\begin{aligned} n &= \text{number of nodes} \\ n_{ij} &= \text{count of nodes labeled i in C and j in A} \\ n_{i.} &= \text{count of nodes labeled i in C} \\ n_{.j} &= \text{count of nodes labeled j in A} \end{aligned}$$

The CRI measure is then:

$$\frac{\sum_{i,j}\binom{n_{ij}}{2} - \sum_i\binom{n_{i.}}{2}\sum_j\binom{n_{.j}}{2} / \binom{n}{2}}{\frac{1}{2}\left[\sum_i\binom{n_{i.}}{2}+\sum_j\binom{n_{.j}}{2}\right] - \sum_i\binom{n_{i.}}{2}\sum_j\binom{n_{.j}}{2} / \binom{n}{2}}$$

4.3. Insufficient Model Complexity: Single Role Label v Multiple Role Labels

If there are in fact C role catagories with 4 possible labels in each, the expectation would be that the effective number of roles in the graph would be 4^C, and therefore any model which uses only 4 labels to model the graph would perform poorly. Figure 3 demonstrates that this is in fact the case – a model with only 4 labels rarely matches any one of the C original role catagories.

The multiple roles labeling algorithm is also expected to generate two catagories of roles which match the two actual role catagories individually. Figure 3 is an also illustration that this property holds – each learned role catagory matches one of the true role catagories very closely (though it is not shown in the chart, they each match distinct role catagories from the actual graph).

4.4. Efficient Data Use: Multiple Roles vs Flat Labels

The underlying role structure can be better modeled with a single "flattened" label. In this case, if our graph has 2 role catagories with 3 values in each catagory, the single role labeler is allowed to use the full 9 role labels. When the graph is large this enables the model to compete with the multiple, however, when the graph is small the multiple roles model still has an advantage, as shown in figure 3. This stems from the fact that the multiple roles model has fewer parameters and reuses the data in the graph more efficiently.

4.5. Asymmetric Label Counts

In some cases, the fact that there is more than one label on each node can cause problems with the Gibbs sampling algorithm. Take a case in which there are 2 role catagories, with 3 elements in the first catagory, and 2 in the second. The algorithm simply assigns two labels to each node – it does not care whether a label is in the first catagory or the second. It may, therefore, model the 3 element catagory with 2 elements, and vice versa. In the experiments in figure 4 56 out of 100 trials had this type of problem. In most cases, the overall accuracy of the labeling is still moderately good, though the matches to individual labels are not as good.

However, one can do better. If the number of labels given to the algorithm for each role catagory is identical, the role catagories once again become truly interchangeable, and it does not matter which gets mapped to which true role catagory. In the experiments shown here, the Gibbs sampler was given two role catagories, each of which had the maximum number of labels across true role catagories (in the case of our graph with 3 and 2 labels in each catagory, the labeler would be asked to find two catagories with 3 labels in each). This model improved accuracy over all 100 trials, as we cans see in the first set of bars in figure 4. The surprising result here was that it does not cause a noticable drop in accuracy when the original results were correct. When the original algorithm would have mapped the 3 role catagory to 3 roles, and the 2 role catagory to 2 roles, resulting in very high accuracy, the model with 3 roles in each catagory barely changes the results (see the third set of bars in figure 4). The key here is that typically the algorithm almost completely eliminates the "extra" label in one catagory, which leads to a very close match in CRI values.

5. Discussion and Future Work

5.1. Finding Information in Real Data Sets

Role labels provide an interesting summary of the behavior of each node, which can then be treated as an attribute and used in other tasks such as classification. Using multiple roles provides an second level of complexity which gives us additional accuracy and information.

If part of the role label for a node is known (e.g., career but not volunteer postition), it becomes easier not just to infer the values of the other role, but to infer the existance of such a role. The model will have a more accurate labeling for careers when the role for volunteer position is added.

5.2. Alternate Inference Procedures

The model used here can be thought of as a bayes net consisting of many repeats of the structure in figure 5, in which two nodes and a single link variable are shown. "Link A" is true if a link is generated by role catagory A, "Link B" is true for links generated by role catagory B, and "link" is the or of these two values. Approximate inference procedures such as mean field, loopy belief propagation, generalized belief propagation should be compared to the sampling algorithm used here.

5.3. Interacting Roles

Independent link generation is convienient, since it leads to a simple model, however, in general, in may not be the case that roles generate links independently.

One interesting example of this type of role interaction is geographical location. Residents of the same town or neighborhood are likely to link to each other – they may attend the same schools, use the same dentist, or run into each other at the supermarket. A dentist in one town looks much like a dentist in another town, however, and still has links to the same other types – hygenists and patients. Geographical location does not so much generate additional links as determine which hygenists and patients the dentist will link to – those with similar locations.

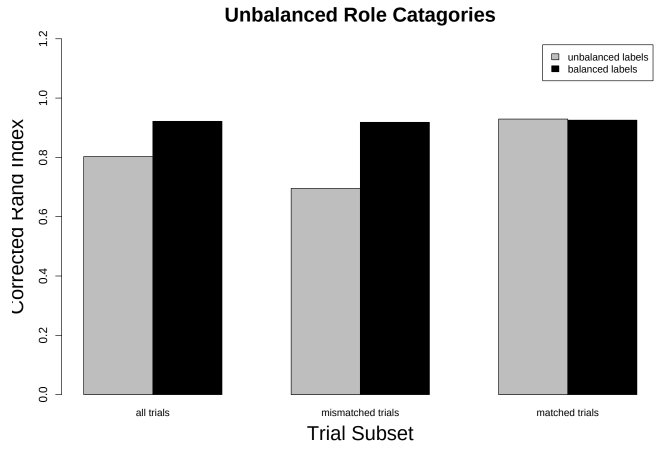

Figure 4. Trials on a graph with 3 roles in one catagory, 2 roles in the other. Two models were learned: an unbalanced model which started out with catagories with 2 and 3 roles, and a balanced model in which both catagories stared out with 3 roles. The leftmost graph plots CRI values for all 100 trials, the second only those trials where the unbalanced model found the wrong mapping of role catagories, and the third only those trials where the unbalanced model found the correct mapping of role catagories.

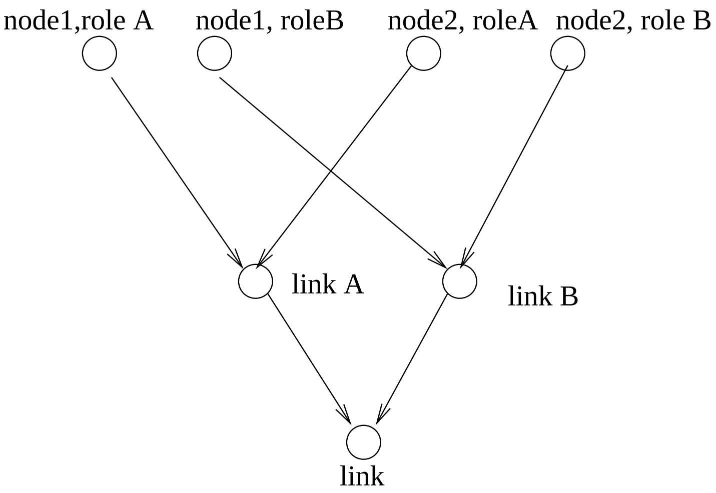

Figure 5. A partial bayes net showing link generation between two nodes, each with labels from two role catagories. The labels in each role catagory independently generate “Link A” and “Link B”. The presence of a link is determined by the “or” of these values.

5.4. Adding in Attribute Information on Nodes and Links

The model can easily be extended to include labeled links, by placing each link type in a separate graph and combining the likelihood of entity labels across these graphs.

Any attribute information which has only local effects (to the node or link on which it is located) can simply be added to the distribution over labels for the node. More complex interactions can introduced into the model, however, once the complexity reaches a certain point it may make more sense to treat the role labels as attribute inputs to another algorithm.

The role label on a node provides summary information about the link pattern for the node. It would be interesting to see if this summary information could at least partially replace or augment the "aggregation" functions used in some models. Aggregation functions compute a mathematical summary of a variable which exists on multiple neighbors of a node (using ave, mode, etc). Certainly the current role model at a first glance seems well suited to replacing aggregators which compute the count of related nodes with particular labels, and could perhaps be extended to compute summaries of other variables.

6. Acknowledgements

This research was facilitated in part by a National Physical Science Consortium Fellowship and by support from Sandia National Laboratories. This research was also supported by DARPA and AFRL under contract numbers F30602-00-2-0597. The U.S. Government is authorized to reproduce and distribute reprints for governmental purposes notwithstanding any copyright notation hereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements either expressed or implied, of DARPA, AFRL, or the U.S. Government.

References

[1] C. Anderson, S. Wasserman, and K. Faust. Building stochastic blockmodels. *Social Networks*, 14:137–161, 1992.

[2] C. J. Anderson, S. Wasserman, and B. Crouch. A p* primer: logit models for social networks. *Social Networks*, 21:37–66, 1999.

[3] V. Batagelj. Notes on blockmodeling. *Social Networks*, 19:143–155, 1997.

[4] Vladimir Batagelj, Anuska Ferligoj, and Patrick Doreian. Generalized blockmodeling. *Informatica (Slovenia)*, 23(4), 1999.

[5] R.L. Breiger, S.A. Boorman, and P. Arabie. An algorithm for clustering relational data with applications to social network analysis and comparison to multidimensional scaling. *Journal of Mathematical Psychology*, 12:328–383, 1975.

[6] Martin G Everett. Role similarity and complexity in social graphs. *Social Networks*, 7:353–359, 1985.

[7] S.E. Fienberg and S. Wasserman. Categorical data analysis of single sociometric relations. *Sociological Methodology*, 12:156–192, 1981.

[8] O. Frank and D. Strauss. Markov random graphs. *Journal of the American Statistical Association*, 81:832–842, 1986.

[9] P. Holland, K.B. Laskey, and S. Leinhardt. Stochastic blockmodels: Some first steps. *Social Networks*, 5:109 – 137, 1983.

[10] P.W. Holland and S. Leinhardt. An exponential family of probability distributions for directed graphs. *JOurnal of the American Statistical Association*, 76(373):33–50, 1981.

[11] L. J. Hubert and P. Arabie. Comparing partitions. *Journal of Classification*, 2:193–218, 1985.

[12] J. Kemeny and J. L. Snell. *Finite Markov Chains*. The University Series in Undergraduate Mathematics. Van Nostrand, 1960.

[13] F. Lorrain and H. C. White. The structural equivalence of individuals in social networks. *Journal of Mathematical Sociology*, 1:49–80, 1971.

[14] Krzysztof Nowicki and Tom A. B. Snijders. Estimation and prediction for stochastic blockstructures. *Journal of the American Statistical Association*, 96(455):1077–??, 2001.

[15] W. M. Rand. Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical Association*, 66:846–850, 1971.

[16] T. Snijders and K. Nowicki. Estimation and prediction for stochastic blockmodels for graphs with latent block structure. *Journal of Classification*, 14:75–100, 1997.

[17] M.G. Everett S.P. Borgatti. The class of all regular equivalences: Algebraic structure and computation. *Social Networks*, 11:65–88, 1989.

[18] S. Wasserman and C. Anderson. Stochastic a posteriori blockmodels: Construction and assessment. *Social Networks*, 9:1–36, 1987.

[19] S. Wasserman and K. Faust. *Social Network Analysis: Methods and Applications.* Cambridge University Press, 1994.

[20] S. Wasserman and P. Pattison. Logit models and logistic regression for social networks: I. an introduction to markov graphs and p*. *Psychometrika*, 61:401–425, 1996.

[21] D. R. White and K. P. Reitz. Graph and semigroup homomorphisms on networks of relations. *Social Networks*, 5:193–234, 1983.